

Université des Sciences et Technologies de Lille
Faculté des Sciences Economiques et Sociales

L'ECONOMIE DU LOGICIEL

Tome I

DE L'ECONOMIE DE L'INFORMATIQUE

A L'ECONOMIE DU LOGICIEL

Thèse de Doctorat nouveau régime
Economie Industrielle

Présentée et soutenue publiquement par

François Horn

Le 18 décembre 2000

Directeur de thèse :

M. Jean Gadrey, Professeur à l'Université de Lille I

Membres du jury :

M. Michel Callon, Professeur à l'Ecole des Mines de Paris

Mme Elisabeth Fichez, Professeur à l'Université Lille III

M. Pascal Petit, Directeur de recherches au CEPREMAP

M. Robert Salais, Directeur de recherches à l'ENS Cachan, rapporteur

M. Olivier Weinstein, Professeur à l'Université Paris XIII, rapporteur

*A Armelle,
Louise, Lucie et Mathilde*

REMERCIEMENTS

Cette thèse n'aurait pas vu le jour sans la très grande disponibilité et les précieux conseils de Jean Gadrey et des membres du groupe " Services, évaluation, innovations " du Centre Lillois d'Etudes et de Recherches Sociologiques et Economiques (CLERSE), sans l'aide multiforme de Bénédicte El Kaladi (dans de très nombreux domaines), de Séverine Costenoble, de Jacques Morelle, de Joachim Da Cunha et de Kourosch Saljoghi, sans le soutien matériel de l'IFRESI et de l'université Lille III, et sans l'indispensable soutien psychologique et affectif de ma famille et de mes proches. Mes remerciements également à tous ceux qui m'ont fait part de commentaires et de remarques sur des parties de ce travail, et notamment à William J. Baumol, Michel Callon, Gérard Dréan et Jean-Benoît Zimmermann.

INTRODUCTION

L'économie du logiciel fait de plus en plus les titres de l'actualité. Si l'on se limite à l'année écoulée, on peut citer sans prétention à l'exhaustivité : le jugement, suite à une poursuite intentée par le département de la justice des Etats-Unis et dix-neuf Etats co-plaignants, condamnant Microsoft à être scindée en deux entités, jugement qui est actuellement en phase d'appel ; les craintes suscitées par le bogue de l'an 2000 ; les succès de certaines *start-up* productrices de logiciels, qui, pour certains commentateurs, sont les fleurons d'une "nouvelle économie" quelque peu mythique ; la chute boursière rapide de nombre de ces nouvelles sociétés ; la réussite étonnante d'un logiciel développé grâce à la coopération bénévole de milliers de programmeurs dispersés aux quatre coins de la planète, le système d'exploitation *Linux* qui est utilisé par plus de 25 % des serveurs Internet ; la création d'un logiciel (*Napster*) qui a permis à près de trente millions d'utilisateurs d'échanger librement des morceaux de musique ; l'apparition du virus *Iloveyou*, qui selon les estimations de *Computer Economics* a infesté en quelques jours trois millions d'ordinateurs dans le monde, avec des dégâts dont le coût est estimé à cinq milliards de dollars.

La fréquence de ces événements et leur couverture médiatique peuvent s'expliquer par le fait que les logiciels jouent un rôle de plus en plus important au sein des technologies de l'information et de la communication, et de ce fait dans l'ensemble de la vie économique et sociale. En effet, les logiciels occupent une place croissante au sein des technologies de l'information et de la communication. Et l'importance de ces technologies ne peut qu'augmenter dans des sociétés où l'information et les connaissances sont de plus en plus déterminantes. Des expressions comme *société de l'information* ou *économie fondée sur les connaissances* désignent l'expansion de biens et services informationnels, mais également le rôle que jouent l'information et les connaissances dans la production de l'ensemble des biens et des services, et dans le fonctionnement de l'économie. Pour certains auteurs, qui ne manquent pas de bons arguments, les technologies de l'information et de la communication seraient d'ailleurs à la base d'un nouveau paradigme technico-économique.

Ce que révèle également la diversité des faits marquants de l'actualité de l'économie du logiciel, c'est que celle-ci apparaît comme un *monde de contrastes et de paradoxes*.

L'économie du logiciel, un monde de contrastes

Ce qui frappe en premier lieu l'observateur de ce secteur, c'est que, quelle que soit la dimension étudiée, l'économie du logiciel présente des caractéristiques apparemment contradictoires. L'économie du logiciel est le lieu où se sont bâties les plus grandes fortunes lors de la dernière décennie¹, mais elle peut être également le siège d'une intense activité matériellement désintéressée comme l'ont montré les efforts consentis pour développer des logiciels libres. Le logiciel est à la fois le type de produit qui est le plus fréquemment approprié illégalement, et une activité qui génère des taux de profits sans précédents historiques. C'est un secteur qui se singularise par l'importance stratégique que peut très rapidement acquérir une entreprise dont la position ne semble plus pouvoir être remise en cause (IBM hier, Microsoft aujourd'hui), et par l'apparition périodique de nouveaux acteurs dont certains réussissent à jouer un rôle majeur dans l'évolution du secteur. Coexistent dans l'économie du logiciel, de très grandes entreprises dont la puissance sur le marché est gigantesque et une myriade de petites entreprises. D'autres phénomènes apparemment contradictoires peuvent être mentionnés : présence de producteurs généralistes exerçant une activité dans l'ensemble de l'informatique, voire au-delà, et de producteurs fortement spécialisés réalisant l'essentiel de leur chiffre d'affaires par la vente d'un seul type de logiciel, mouvements permanents d'internalisation et d'externalisation entre une production réalisée en interne par les services informatiques des organisations utilisatrices et une production mise sur le marché par des entreprises productrices de logiciels, innovations incessantes et inertie technologique sur certains aspects des logiciels et de leur production. Comme le souligne David C. Mowery, "l'industrie du logiciel est caractérisée par l'extraordinaire dynamisme et la fluidité de ses structures" (1996, p.3).

Le caractère éminemment contrasté des producteurs de logiciels s'explique en partie, mais en partie seulement, par la diversification croissante des logiciels et de leurs utilisateurs. Concernant les produits, la diversité fonctionnelle des logiciels augmente continuellement avec l'extension des domaines d'application de l'informatique et la création permanente de nouveaux types de produits. Les produits réalisés sont extrêmement divers, depuis les

¹ Le titre d'homme le plus riche du monde est détenu alternativement ces derniers temps par deux entrepreneurs de l'économie du logiciel, Bill Gates de Microsoft et Larry Ellison d'Oracle. Le classement entre ces deux hommes se modifie fréquemment en raison des évolutions rapides de la valeur boursière de leurs sociétés, la valeur de l'action Microsoft fluctuant fortement suivant les rebondissements du procès Microsoft.

quelques lignes de programme écrites par un utilisateur pour ses besoins propres, jusqu'aux logiciels de contrôle de la navette spatiale, en passant par des produits aussi différents que des jeux, des traitements de texte, des systèmes de gestion d'entreprise... A cette diversité technologique, se superpose la diversité des logiciels sur un plan économique, avec l'existence d'un continuum de situations allant de prestations complètement réalisées sur mesure, jusqu'à l'existence de produits standard (progiciels) vendus à des millions d'exemplaires. Concernant les utilisateurs, on remarque leur extrême hétérogénéité, y compris pour un logiciel d'un type donné. En particulier, si pour la majorité des utilisateurs, les rapports avec l'informatique s'apparentent à un banal usage d'outils qui étend leurs capacités de création, d'obtention et de traitement de l'information dans un domaine particulier, il existe également des utilisateurs pour lesquels un logiciel est l'objet d'une véritable pratique, chargée d'émotions et de passions, où intervient le système de valeurs propres de l'utilisateur.

Le paradoxe de l'économie du logiciel.

La deuxième considération initiale est que l'économie des logiciels semble marquée par un paradoxe majeur. Schématiquement la fonction principale de la plupart des logiciels est d'améliorer la productivité et la qualité de l'activité des utilisateurs. Or, apparemment, l'économie du logiciel, quant à elle, se caractérise par la faiblesse des gains de productivité pour produire des logiciels (que ce soit en termes de niveau ou d'évolution) et par une qualité insuffisante de la majorité des produits créés, alors même qu'un logiciel est lui-même produit en grande partie à l'aide d'autres logiciels. Comment expliquer qu'un secteur dont l'objectif est d'améliorer les performances des autres secteurs soit lui-même, au moins en première analyse, si peu performant ? N'est-il pas surprenant que la production d'un outil destiné à rationaliser les activités, soit elle-même une activité si difficile à rationaliser ?

Concernant la faiblesse des gains de productivité dans la production des logiciels, William J. Baumol, Sue Anne Batey Blackman et Edward N. Wolf en déduisent que le secteur des logiciels constitue un exemple typique d'un "secteur stagnant", responsable de la "croissance asymptotiquement stagnante" de l'ensemble de l'informatique (1985, p. 813). C'est à partir d'un constat identique qu'une étude américaine, citée par l'OCDE (1991 A) concluait que si les tendances actuelles se poursuivaient, en 2040, toute la population des Etats-Unis (hommes, femmes et enfants) devrait écrire des logiciels !

Concernant la qualité des logiciels produits, le manque de fiabilité des logiciels (résultant soit d'une erreur de comportement du programme dans une situation donnée, soit

d'une situation non prévue), et la relative inadaptation aux besoins des utilisateurs suscitent un mécontentement croissant avec l'importance prise par les logiciels dans la vie économique et sociale. Il apparaît en effet surprenant que des produits de haute technologie comme les logiciels puissent comporter tant de défauts, dont certains apparaissent tellement évidents une fois qu'ils ont été découverts, et induire des comportements parfois si déconcertants de la part des systèmes informatiques qu'ils pilotent. S'il importe d'analyser précisément l'ampleur réelle de ces phénomènes, il est indéniable qu'ils constituent des questions critiques de l'économie des logiciels, dans un secteur où la croissance des besoins, quantitatifs et qualitatifs, est particulièrement rapide.

Le logiciel, un objet déconcertant pour l'économiste.

Si le logiciel est à bien des égards un objet particulièrement déconcertant pour ses utilisateurs, il l'est peut-être encore davantage pour l'économiste. Le logiciel est défini, comme "l'ensemble des programmes, procédés et règles et éventuellement la documentation, relatifs au fonctionnement d'un ensemble de traitement de données"². Mais au-delà de cette définition, qu'est-ce, économiquement, qu'un logiciel ? S'agit-il d'un bien ou d'un service ? Est-ce un bien rare, dans le sens que la science économique donne à ce qualificatif quand elle se définit comme la science de la gestion des ressources rares ? Possède-t-il les caractéristiques des biens collectifs ? Est-ce un produit sur lequel peuvent s'exercer des droits de propriété ? A quelles conditions techniques et/ou juridiques ? Qu'est-ce qui détermine son prix ? Les théories de la valeur des économistes peuvent-elles lui être appliquées ?

Curieusement, l'ampleur de ces questions a suscité peu de recherches et de travaux économiques concernant spécifiquement les logiciels³. Comme le note David C. Mowery, éditeur de l'un des rares ouvrages consacrés à l'économie du logiciel (*The international computer software industry. A Comparative Study of Industry Evolution and Structure*) "l'industrie du logiciel a peu attiré l'attention de manière surprenante, étant donné sa taille, sa croissance rapide et son importance de plus en plus grande dans les industries de haute

² Cette définition est extraite d'un arrêté du 22 décembre 1981, publié au Journal Officiel du 17 Janvier 1982 relatif à l'enrichissement du vocabulaire français. Elle est très proche de celle contenue dans la norme ISO relative au logiciel : "le logiciel est un ensemble de programmes de calculateur, procédés, règles et éventuellement de la documentation associée, relatif au fonctionnement d'un système de traitement de l'information". Serge Bouchy fait observer que "les définitions les plus récentes du logiciel le décrivent comme étant l'association de la documentation, du logiciel proprement dit et de ses SIC (systèmes d'information de contrôle ou tests) de validation" (1994, p. 177).

technologie" (1996, p. 312). L'analyse de l'économie du logiciel se limite le plus souvent à un "sous-produit" d'une analyse de l'économie de l'informatique, centrée sur les produits matériels informatiques et qui a suscité davantage de recherches. Certes l'étude des logiciels ne peut être menée indépendamment de celle des systèmes informatiques dont ils sont un des composants. Mais, si une analyse de l'économie du logiciel ne peut faire l'impasse sur une analyse économique de l'évolution de l'informatique dans laquelle elle s'insère, elle ne peut se limiter à cela. D'une part, les logiciels présentent suffisamment de traits originaux, de différences avec les matériels informatiques pour justifier une analyse économique spécifique. D'autre part, si, à l'origine, l'économie du logiciel occupait une place assez marginale au sein de l'économie de l'informatique, c'est de moins en moins le cas actuellement. De ce point de vue, il est hautement significatif que les questions de standardisation, qui ont toujours joué un rôle majeur dans l'informatique, se soient progressivement déplacées des questions matérielles vers les questions logicielles.

Par contre, la littérature scientifique et technique, qui traite des logiciels, est riche et variée, et il existe de nombreuses publications (professionnelles et grand public). L'éclosion de nombreux bulletins et lettres d'informations *on line*, qui leur consacrent fréquemment une large place, a encore augmenté l'abondance de l'information en la matière. Nous avons largement utilisé ces sources. Elles fournissent des éclairages intéressants sur les spécificités techniques des logiciels et les particularités de cette activité. La prise en compte de ces caractéristiques techniques spécifiques est certainement indispensable pour comprendre pourquoi il est impossible de construire des logiciels sans défaut, pourquoi l'intercompréhension entre les concepteurs et les utilisateurs des logiciels est si malaisée, ou pourquoi il est si difficile d'atteindre une productivité élevée dans le développement d'un logiciel mais aussi pourquoi c'est une activité qui peut susciter tant de passion.

Si ces apports sont indispensables, ils ne sont pas suffisants pour analyser en profondeur les phénomènes observés. Il est nécessaire de faire la part entre ce qui est imputable aux spécificités de l'activité et du produit, et ce qui relève de déterminants plus "économiques". De ce point de vue, une analyse économique peut constituer une contribution particulièrement intéressante, même si elle suscite parfois un certain scepticisme de la part des spécialistes des logiciels. Par exemple, Capers Jones, fondateur et président de la société *Software*

³ Une explication possible est que, de façon générale, l'attention des économistes s'est longtemps essentiellement focalisée sur les biens industriels.

Productivity Research, dans un livre intitulé "*Programming Productivity*", estime que "le manque de mesures économiques satisfaisantes pour le logiciel a commencé à susciter l'idée que l'économie elle-même n'était pas une science complète et que certaines familles de produits, tels que les systèmes logiciels, n'étaient entièrement couverts ni par les théories, ni par les modèles économiques" (1989, p. 108). Nous ne partageons pas totalement ce pessimisme concernant les théories économiques. De multiples champs des sciences économiques peuvent être utilement mobilisés pour l'analyse des logiciels. Ils peuvent saisir leur nature de produits informationnels soumis à de fréquentes innovations : économie de l'information et de la connaissance, économie industrielle et économie des services, économie du changement technique. En particulier un certain nombre d'outils économiques ont un pouvoir heuristique indéniable pour l'analyse de l'économie du logiciel. Nous utiliserons notamment les notions d'externalités de réseaux directes et indirectes, de rendements croissants d'adoption, d'effets "feedback", de situations de verrouillage technologique, de biens collectifs, de biens intangibles, ainsi que des analyses en termes de caractéristiques des produits, les approches des différents types d'innovations, et de la distinction entre les effets médiats et immédiats d'une activité...

Plus généralement, les apports d'une analyse économique de notre objet d'étude nous semblent d'autant plus féconds qu'ils ne se limitent pas à une vision strictement "économiste" des phénomènes. Il en est ainsi de l'analyse de l'évolution de l'ensemble de l'informatique, dans lequel s'intègre l'économie du logiciel, qui peut être enrichie par l'utilisation de la notion de réseau technico-économique développée par Michel Callon (1991). Concernant plus précisément l'économie du logiciel, le véritable défi consiste, dans ce travail, à trouver un cadre d'analyse qui permette d'expliquer en quoi et pourquoi l'économie du logiciel est un *monde de contrastes et de paradoxes*. Robert Salais et Michael Storper (1993) ont construit une typologie permettant d'appréhender, au niveau macro-économique, la pluralité des modèles de systèmes productifs qui peuvent coexister au sein d'une économie nationale. Notre projet, dans cette thèse, est de mettre en place un cadre d'analyse de même nature, que l'on peut qualifier de socio-économique en raison de ses dimensions sociales et historiques. Ce cadre théorique nous permet d'intégrer à l'analyse de l'économie du logiciel, la pluralité des registres d'action qui dépassent le recours au seul calcul économique privé, et la diversité des mécanismes de coordination qui intègrent l'existence de conventions et d'institutions.

Notre démarche comporte deux grands stades d'analyse de l'économie du logiciel, qui correspondent aux deux parties de la thèse. La première partie ("*De l'économie de l'informatique à l'économie du logiciel*") permet de cerner la réalité de l'économie du logiciel à partir de l'économie de l'informatique dont elle est un des éléments constitutifs.

Il s'agit tout d'abord d'essayer de comprendre ce qu'est un logiciel, son rôle, son processus de production, ses caractéristiques pertinentes. Cela nécessite de situer les logiciels dans l'étude des systèmes informatiques qu'ils pilotent. Un système informatique a pour fonction, de façon très générale de traiter de l'information. Une des premières difficultés consiste à cerner la notion d'information, et notamment à la distinguer de notions voisines comme celles de données et de connaissances. Ceci nous permettra d'étudier les particularités du traitement automatique de l'information, et le statut spécifique des logiciels dans ce processus. Enfin, ce premier chapitre analyse la production d'un logiciel comme étant un travail intellectuel de production de connaissances codifiées, et étudie les différentes caractéristiques des logiciels et leur articulation.

Le deuxième chapitre a pour fonction de situer l'évolution de l'économie du logiciel dans l'histoire de l'informatique. Pour ce faire, il est nécessaire de prendre en compte *simultanément* les innovations techniques, les comportements stratégiques des producteurs mais aussi les changements concernant la nature des principaux utilisateurs, les types de problèmes qu'ils souhaitent résoudre, et les modifications de leurs attentes, liées à des évolutions culturelles et sociales plus générales. Or, si chacun de ces aspects a fait l'objet de recherches technologiques, économiques, sociologiques séparées, il n'existe pas, à notre connaissance, de tentative d'intégration dans une approche unifiée de l'ensemble de ces dimensions, ce que nous nous proposons de faire à partir de la notion de réseau technico-économique. L'objectif n'est pas seulement de comprendre les évolutions passées mais également de cerner les enjeux essentiels des changements structurels actuels et d'esquisser les perspectives d'évolution probable.

Le troisième chapitre analyse pourquoi le logiciel est un produit particulièrement difficile à appréhender par l'économiste. Il l'est tout d'abord en raison de son extraordinaire diversité fonctionnelle. Les logiciels pénètrent les domaines d'activité les plus variés et sont confrontés à des problèmes de plus en plus complexes, auxquels il semble que l'informatisation, grâce à l'augmentation de la puissance des matériels et à la baisse de leur coût, puisse apporter une solution. En même temps, dans certains cas, l'informatisation d'une

situation peut contribuer à la complexifier davantage. Dans l'économie des logiciels se produisent de multiples innovations qui concernent l'ensemble des dimensions de la production des logiciels : apparition de nouveaux produits, de nouvelles méthodes de production, utilisation de nouveaux outils. Si parfois les changements peuvent être extrêmement rapides, il faut également prendre en compte le temps et les efforts d'assimilation et d'apprentissage, qui expliquent que la diffusion de certaines innovations puisse être beaucoup plus lente. Ces évolutions qui se produisent sur une période très courte (la production de logiciel comme activité distincte n'a pas plus de trente années d'existence) expliquent l'instabilité technologique qui caractérise l'économie du logiciel, et rendent plus difficile son analyse. Sur un plan économique le logiciel n'est pas plus simple à analyser. C'est un produit hétérogène occupant des positions diverses sur le continuum des biens et des services. Parmi les logiciels, les progiciels possèdent "naturellement" les déterminants des biens collectifs, et il est nécessaire de recourir à des mécanismes techniques et juridiques sophistiqués pour contrôler leur reproduction et en faire des produits commercialisables. Enfin, l'évolution des prix des logiciels fait intervenir de multiples mécanismes et il est difficile de leur appliquer les théories habituelles de la valeur de la science économique.

Le quatrième chapitre analyse l'importance des logiciels dans l'économie. On y étudie tout d'abord la place prise par les technologies de l'information et de la communication, dont font partie les logiciels, et la convergence éventuelle de ces technologies, permise par la numérisation des contenus. La question de savoir si ces technologies sont à la base d'un nouveau paradigme est examinée en confrontant les différentes significations que peut prendre la notion de paradigme appliquée à l'économie, et les principales tendances observées dans le fonctionnement de l'économie. La deuxième partie de ce chapitre tente d'apprécier, malgré les difficultés statistiques, l'importance quantitative et qualitative prise par les logiciels au sein de ces technologies, ce qui conduit à mettre en évidence l'existence d'un processus que l'on peut qualifier de "logicialisation" de l'activité économique et sociale.

La deuxième partie (*"De l'économie du logiciel à la socio-économie des « mondes de production » des logiciels"*) examine tout d'abord les deux particularités fondamentales de l'économie du logiciel avant d'en proposer un cadre explicatif commun, qui permette également d'appréhender la dynamique de l'économie du logiciel.

La première de ces particularités, traitée dans le cinquième chapitre, est la permanence des "questions critiques" de l'économie du logiciel. L'économie du logiciel fait l'objet depuis

ses origines de trois critiques récurrentes : un manque de fiabilité des logiciels produits, une relative inadaptation aux besoins des utilisateurs, et une évolution jugée insuffisante de la productivité dans la production des logiciels, avec ses conséquences en termes de coûts et de délais. Les deux premiers aspects expliquent l'importance démesurée prise par la maintenance, qu'il s'agisse d'une maintenance corrective destinée à éliminer les imperfections constatées ou d'une maintenance adaptative pour améliorer l'adéquation du produit aux attentes des utilisateurs, qui à son tour influe négativement sur la productivité globale du secteur. La réalité de ces problèmes est attestée par la perception des utilisateurs et par l'importance qu'ils occupent dans la littérature spécialisée, qui fourmille d'exemples spectaculaires. Nous tenterons d'effectuer une analyse précise de l'ampleur réelle de ces phénomènes. La persistance de ces problèmes ne signifie pas qu'il n'existe pas des progrès réels dans la production des logiciels, mais plus vraisemblablement que ces progrès restent insuffisants par rapport à l'augmentation de la quantité et à la diversité des logiciels à produire, aux exigences croissantes des utilisateurs, et à la complexité grandissante de ce qui est informatisé.

La deuxième particularité fondamentale de l'économie du logiciel, analysée dans le sixième chapitre, est la diversification croissante de l'économie du logiciel. Celle-ci est examinée sous l'angle de la diversité des produits et des utilisateurs, et surtout sous l'angle de la diversité des producteurs. La diversité des producteurs est appréhendée selon trois dimensions classiques de l'économie industrielle : le statut des producteurs qui recouvre les oppositions entre *faire* et *faire faire*, la taille des entreprises avec les phénomènes de concentration et de multiplication des entreprises due à l'apparition de nouveaux producteurs, l'activité principale des producteurs avec les oppositions entre l'intégration et la désintégration, et entre la diversification et la spécialisation.

Le septième chapitre présente tout d'abord le cadre de la théorie des mondes de production en réexaminant certaines des questions clés de la science économique qui sont à la base de sa construction : analyse de l'incertitude et plus particulièrement de l'incertitude radicale, conception de la rationalité des agents économiques en montrant l'importance des formes "non instrumentales" de la rationalité, analyse des institutions dépassant les explications en termes de défaillance du marché, importance des conventions dans les différentes modalités de coordinations entre agents économiques. Moyennant quelques adaptations la typologie des mondes de production de Robert Salais et Michael Storper est appliquée à l'économie des logiciels. Elle permet de mettre en évidence les fondements de la

diversité observée dans l'économie du logiciel, à savoir l'existence de conventions différentes, chacune centrée sur un type de produit, en réponse aux questions critiques de l'économie du logiciel.

Le huitième et dernier chapitre vise à montrer que la coexistence de différents mondes de production est probablement un phénomène durable, et non un phénomène transitoire lié à la jeunesse de l'économie du logiciel. L'argumentation est centrée sur l'existence d'importantes relations de complémentarité entre les mondes de production des logiciels. Toutefois la persistance de plusieurs mondes de production des logiciels ne doit pas masquer les profondes évolutions de ce secteur, qui se concrétisent notamment par des changements dans l'importance respective de chacun de ces mondes. Il se produit à l'intérieur et entre les différents mondes de production des évolutions dynamiques, qui sont analysées en identifiant les principales trajectoires à l'œuvre. L'intérêt de cette analyse est également de montrer comment certaines dynamiques ont déjà affecté différemment les trois grandes zones de production des logiciels que sont les Etats-Unis, le Japon et l'Europe, et de mettre en évidence les perspectives d'évolution les plus souhaitables – mais pas nécessairement les plus probables – pour l'ensemble de l'économie du logiciel, ce qui peut motiver des actions volontaristes.

Chapitre I

DU TRAITEMENT AUTOMATIQUE DE L'INFORMATION

AUX CARACTERISTIQUES DES LOGICIELS

Pour comprendre ce qu'est un logiciel, son rôle, son processus de production, ses caractéristiques pertinentes, il est nécessaire de le situer dans un système informatique dont il est un des composants. Un système informatique a pour fonction, de façon très générale, de traiter de l'information. Une des premières difficultés consiste à cerner la notion d'informations, et notamment à la délimiter par rapport à des notions voisines comme celles de données et de connaissances. C'est ce que nous tenterons de faire dans un premier temps, en examinant également les implications de ces précisions sur l'analyse de l'efficacité du processus informationnel et sur l'analyse économique de l'information (section I). Dans une deuxième étape, nous étudierons les particularités du traitement automatique de l'information (section II), ce qui nous permettra de définir un logiciel et d'analyser son statut (section III). Nous verrons ensuite en quoi la production d'un logiciel est un travail intellectuel de production de connaissances codifiées (section IV), avant d'étudier les différentes caractéristiques des logiciels et leur articulation (section V).

Section I - Données, informations, connaissances : une tentative de clarification

En France, le terme informatique a été créé en 1962 par Pierre Dreyfus par la fusion des termes information et électronique. Sa définition, reprise par l'Académie Française en 1967, est la suivante : "science du traitement rationnel, notamment à l'aide de machines automatiques, de l'information, considérée comme le support de connaissances dans les domaines scientifique, économique et social" (Jean-Yvon Birrien, 1990, p. 3). Les Anglo-saxons se limitent à l'expression *data processing*, ce qui nourrit une première interrogation : que traite un ordinateur (*electronic data processing*) ? Des informations ou des données ? Les

ambitions plus récentes de l'intelligence artificielle d'en faire des machines à traiter des connaissances ont encore complexifié cette question. Quelles distinctions faut-il introduire entre les notions de données, d'informations et de connaissances ? Paradoxalement, l'importance acquise par ces trois notions, qui se reflète dans le succès des expressions "bases et banques de données", "société de l'information", "économie fondée sur la connaissance", "management des connaissances" (*knowledge management*), n'est pas allée de pair avec une clarification de ces notions. Celles-ci semblent condamnées à receler une "importante ambiguïté" (Philip E. Agre, 1997, p.251) et "chaque cadre théorique tend à privilégier un type d'information et de connaissance en le dressant contre les autres" (Laurent Thévenot, 1997, p. 208)¹. Les frontières entre ces trois domaines qui sont variables selon les auteurs² semblent même s'estomper au fur et à mesure du développement de l'informatique : avec le développement du "paradigme numérique" (Godefroy Dang Nguyen et Denis Phan, 1997, p. 184), toutes les informations quelle que soit leur forme (écrits, sons, images...) reposent sur le même type de données élémentaires (des bits) ; les progrès dans la codification des connaissances, rendue plus attractive par l'utilisation des ordinateurs, permettent de générer des masses croissantes d'informations, qui occupent elles-mêmes une place de plus en plus importante dans les processus d'acquisition des connaissances...

Une réponse possible, énoncée par certains auteurs, consiste à élargir la portée du concept d'information, tout en distinguant différents types : information-structure, information-méthode et information-ressource dans l'analyse d'Anne Mayère (1990), information-support et information-service dans l'analyse de Thierry Ribault (1993). En dépit de l'intérêt de ces analyses pour appréhender la production de l'information (Anne Mayère) ou sa nécessaire gestion patrimoniale (Thierry Ribault), ces distinctions semblent peu opératoires pour étudier les spécificités du traitement automatique de l'information et le rôle des logiciels, et il semble plus opportun de restreindre le champ conceptuel de l'information en maintenant une distinction avec les données et les connaissances. Ceci nécessite de définir précisément

¹ Jacques Printz souligne que "on ne sait pas très bien parler de l'information, et encore moins la définir. De même avec la complexité ou la sémantique (...). On ne sait pas non plus les mesurer (...) Aucune comparaison possible avec la matière, l'énergie ou les ondes, qui nous sont relativement familières depuis plusieurs siècles" (1998, p. 14).

² Alvin Toffler (1990) indique qu'il utilise indifféremment les termes "donnée", "information" et "connaissance" pour éviter les "répétitions lassantes" !

ces trois notions, tout en étant conscient de la nature relativement conventionnelle des distinctions opérées.

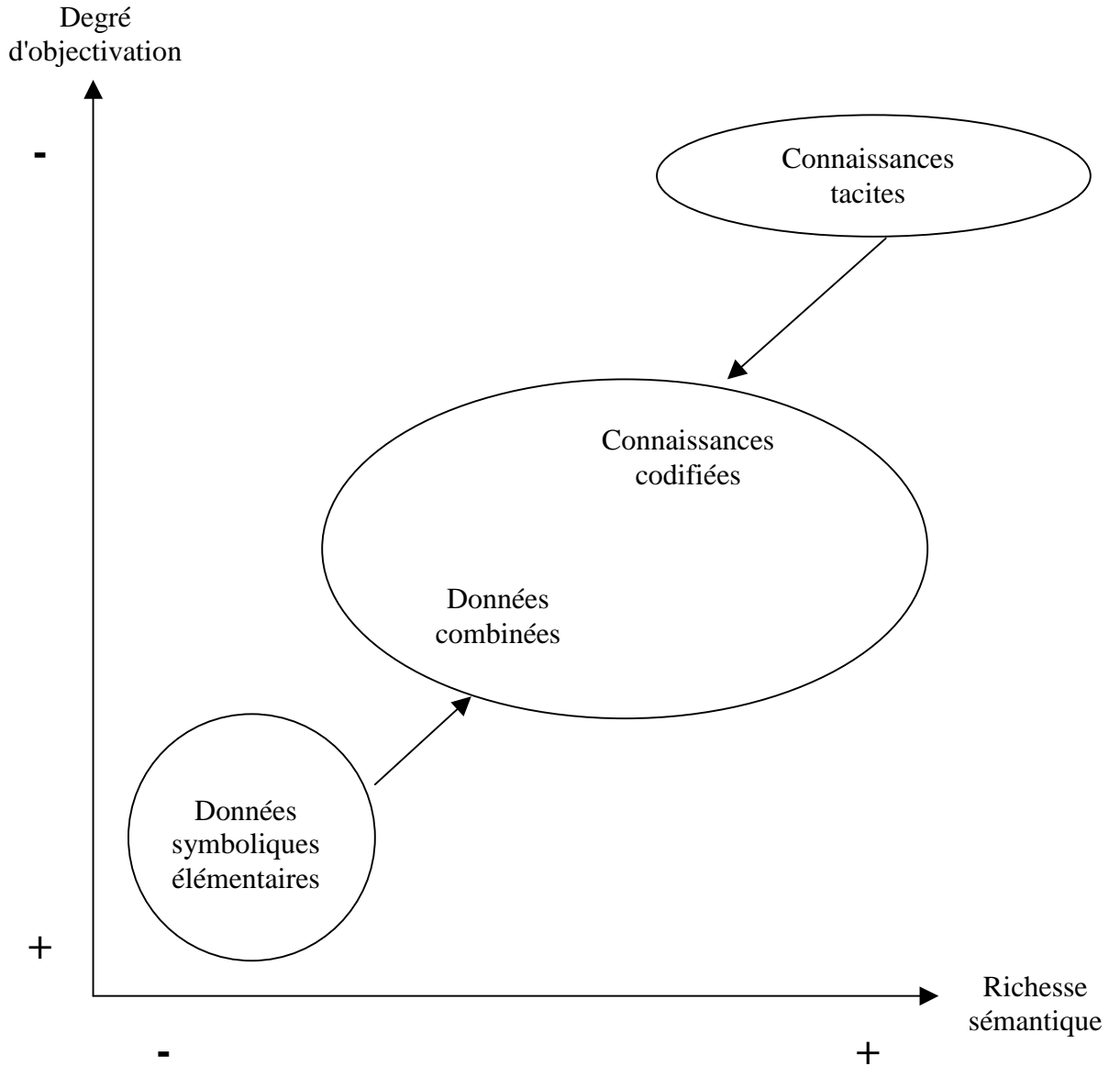
Les données et les connaissances sont tout d'abord définies, en examinant les ruptures et les continuités existant entre elles (A), avant d'explicitier la notion d'information, définie comme l'élément central d'un processus de communications de connaissances (direct ou médié) (B). Ceci nous permettra d'analyser l'efficacité du processus informationnel (C) et d'examiner quelques conséquences sur l'économie de l'information (D).

A - DONNEES ET CONNAISSANCES : RUPTURES ET CONTINUITES

Dans un premier temps, sont opposées de façon polaire les formes extrêmes que peuvent prendre les données (données symboliques élémentaires) et les connaissances (connaissances tacites) selon deux dimensions : le degré d'objectivation et la richesse sémantique³. Dans un second temps, est mise en évidence l'existence d'une zone de continuité autour des notions de données combinées et de connaissances codifiées (cf. schéma I).

³ "La sémantique s'intéresse à la relation qui existe entre le système de signes (syntaxe) et les entités du monde réel que ces signes sont censés représenter" (Jacques Printz, 1998, p. 219).

Schéma I : Données et connaissances



Les données symboliques élémentaires sont diverses : lettre, chiffre, octet, pictogramme. Elles constituent une réalité objective à un double niveau : d'une part, elles sont inscrites sur un support matériel et peuvent circuler indépendamment de leur créateur ; d'autre part leur signification, dépourvue d'ambiguïté, est dépendante d'un code purement conventionnel et dont l'espace de validité peut être extrêmement vaste (universel dans certains cas). Par contre, leur contenu sémantique est limité par le nombre de symboles différents, la

forme extrême étant le *bit*, symbole binaire, qui ne peut avoir que deux significations possibles.

A l'opposé, les connaissances tacites (incorporées) apparaissent comme fortement subjectives, intrinsèques aux individus, complexes et étendues (Anne Mayère, 1997, p. 134)⁴. Ikujiro Nonaka et Hirotaka Takeuchi (1997, p. 26) soulignent que les connaissances tacites sont "profondément enracinées dans l'action et l'expérience d'un individu autant que dans ses idéaux, ses valeurs et ses émotions" et qu'elles possèdent une dimension technique (habiletés et aptitudes concrètes) mais également une importante dimension cognitive ("schémas, modèles mentaux, croyances et perceptions qui sont si enracinés qu'on les considère comme allant de soi"). La subjectivité des connaissances renvoie moins à un niveau individuel qu'au niveau des multiples groupes sociaux auxquels appartient l'individu et qui influencent son action : "ce que nous ne pouvons pas articuler, nous le savons à travers la manière dont nous agissons. Le savoir et le faire ne sont pas séparables (...). Je changerai (...) non pas au nom d'un choix, non pas en fonction de l'acceptation d'une règle évaluée consciemment, non pas au niveau d'un processus conscient, mais parce qu'en faisant comme les autres, je découvrirai que je sais ce qu'ils savent. En sachant ce qu'ils savent, je ferai ce qu'ils font (...). Le lieu du savoir apparaît comme étant non pas l'individu mais le groupe social ; ce que nous sommes en tant qu'individus n'est rien d'autre qu'un symptôme des groupes où se niche le quantum de savoir irréductible" (Harry M. Collins, 1992, p. 19).

Nous considérons la notion de connaissance dans son acception la plus large comme englobant l'ensemble des savoirs (connaissances déclaratives), des savoir-faire (connaissances procédurales), des savoir-être (connaissances comportementales), ayant un contenu sémantique riche et varié. La richesse sémantique des connaissances tacites provient du fait que leurs significations sont multiples, étroitement dépendantes du contexte matériel et humain dans lesquelles elles sont appliquées, et des autres connaissances notamment implicites du détenteur de ces connaissances. En ce sens, on peut considérer que les compétences d'un acteur (interaction dynamique entre les différents types de savoirs) se résument à son stock de connaissances ou capital humain. Ce stock présente toutefois une particularité : à la différence d'un stock de ressources qui s'épuise dans l'usage qui en est fait,

⁴ De façon imagée, Paul Strassman (ancien directeur informatique du Département de la Défense aux Etats-Unis) définit la connaissance comme étant "ce qui quitte chaque soir les bureaux de l'entreprise" (cité par Jean-Marie Desaintquentin et Bernard Sauter, 1996, p. 265).

des connaissances nouvelles peuvent être générées par l'action. Cet acteur peut être individuel ou collectif ; dans ce dernier cas, les compétences de l'organisation peuvent être différentes de la somme des compétences des acteurs individuels qu'elle intègre, de façon négative quand le fonctionnement de l'organisation laisse inemployées les compétences de certains de ses membres⁵, mais aussi de façon positive par la production de connaissances tacites collectives (Ikujiro Nonaka, Hirotaka Takeuchi, 1997), produit de la synergie qui peut exister entre les différentes compétences individuelles⁶.

On peut donc à cette étape différencier les données des connaissances par le fait que le support des premières est un objet physique alors que le support des secondes est un acteur (individuel ou collectif) humain, et par le fait qu'aux significations nécessairement limitées attribuées par un code à des données symboliques élémentaires est opposée l'infinie diversité des connaissances d'acteurs différents.

Toutefois cette opposition est moins tranchée si l'on prend en compte deux phénomènes : l'association de données élémentaires et la codification des connaissances.

L'association de données élémentaires sous forme de mots, de phrases, de textes, de nombres, de séries statistiques, est porteuse d'une signification qui dépasse chacun des éléments qui les composent. Elle pourra exprimer une mesure, un coût, une désignation, un état, etc.. Par contre, cet enrichissement sémantique, permis par un système de codage qualitativement plus riche, s'accompagne d'une objectivation moindre des données, les conventions à la base de la signification de ces associations ayant une portée moins générale (des langages de diffusion plus restreinte, des jargons⁷) et la personnalité du producteur de ces données combinées (la source) acquérant une plus grande importance. Toutefois, dans ce processus, l'output reste séparable du producteur et les données peuvent donc être transmises indépendamment de leur concepteur.

⁵ Interviennent notamment les problèmes de communication : "la compétence collective est supérieure à la somme des compétences individuelles modulo les capacités de communication" (Jacques Printz, 1998, p. 85).

⁶ Pascal Petit parle de "connaissance diluée" pour désigner des situations dans lesquelles la "connaissance ne peut être exhibée que par la mise en commun des savoirs particuliers" (1998, p. 361).

⁷ Si la langue française comporte environ soixante mille mots, les auteurs en utilisant en moyenne six mille, le vocabulaire technique actuel rassemble de l'ordre de six millions de termes (Christine Afriat, Pierre Caspar, 1988, p. 7).

Cette situation qui aboutit à "des formes de connaissances susceptibles d'être abstraites des choses, des personnes et des situations, de se généraliser et de circuler" (Laurent Thévenot, 1997, p.207-208) peut également être appréhendée à partir des connaissances tacites et de leur transformation en connaissances codifiées.

La codification des connaissances.

Les connaissances codifiées (explicites) sont des connaissances énoncées dans un langage formel, inscrites sur un support et "exprimées dans un format standardisé qui les rend compactes, ce qui permet une transmission, une vérification, un stockage et une reproduction aisés et peu coûteux" (Partha Dasgupta, Paul A. David, 1994). La numérisation a prolongé et intensifié le mouvement vers la codification⁸, permettant une expansion cumulative de la base de connaissances codifiée (M. Abramowitz et P. David, 1996) : "la révolution numérique a accentué l'évolution vers la codification du savoir et modifié la part du savoir codifié et du savoir tacite dans le fonds global de savoir de l'économie" (OCDE, 1996, p. 13). Cependant, il faut noter que toute opération de codification des connaissances est également un processus d'appauvrissement du contenu sémantique de celles-ci, par la réduction des savoirs sous la forme de messages faciles à traiter et à diffuser⁹. La codification d'une connaissance tacite, en excluant nécessairement une partie du contexte qui ne pourra être modélisée et la part des compétences non explicitement codifiées, réduit la signification qu'avait cette connaissance, ce que Dominique Foray et Bengt Ake Lundvall nomment "l'incomplétude de la connaissance codifiée" (1997, p. 28). C'est ce qui explique que pour exploiter valablement une connaissance codifiée, il est nécessaire de reconstituer un contexte pertinent à cette connaissance et de disposer des compétences complémentaires nécessaires. Certaines connaissances incorporées (inscrites dans des corps humains ou dans des instruments ou machines) prennent la forme de savoir-faire, de tours de main, d'automatismes techniques,

⁸ Ikujiro Nonaka et Hirotaka Takeuchi (1997) nomment *extériorisation* la conversion des connaissances tacites en connaissances explicites, ce qui évoque un processus d'objectivation des connaissances (des connaissances indépendantes de celui qui sait).

⁹ Michael Polanyi (1966, p.4) soulignait à propos des connaissances tacites que "nous savons toujours plus que ce que nous pouvons dire" et développe un certain nombre d'exemples ("savoir rouler à bicyclette", "savoir reconnaître un visage") pour montrer comment une part de cette connaissance échappe presque naturellement à la description et à l'explicitation. Pour les organisations, Jean-Claude Tarondeau considère que les connaissances codifiées constituent le "niveau bas du savoir" (1998, p. 45). Toutefois ce constat ne doit pas conduire à sous-estimer l'importance de l'activité créative nécessaire pour expliciter les connaissances tacites, nécessitant leur transformation active lors de la modélisation (cf. pour la constitution de systèmes experts, Armand Hatchuel, Benoît Weil, 1992).

dont aucune description ne permet d'épuiser le contenu (Michel Callon, 1993, p. 6). De même Olivier Weinstein (1989) insiste sur l'importance des connaissances tacites dans le produit des activités de recherche-développement, indissociables des outputs codifiés que sont les brevets et publications. En privilégiant les relations de complémentarité par rapport aux relations de substituabilité entre connaissance tacite et connaissance codifiée¹⁰, certains auteurs vont jusqu'à affirmer que la proportion des deux types de connaissances n'a pas profondément changé, toute addition à la base de connaissance codifiée s'accompagnant d'une addition équivalente à la base de connaissance tacite (Robin Cowan, Dominique Foray, 1998, p.314). La connaissance codifiée étant "toujours incomplète, elle suppose par définition la création d'une nouvelle connaissance tacite, [qui] ressurgit sans cesse sous des formes nouvelles" (Dominique Foray, Bengt Ake Lundvall, 1997, p. 28)¹¹.

Au-delà de l'appréciation difficile à évaluer empiriquement de l'évolution de la part relative des connaissances codifiées (problème de la mesure d'une quantité de connaissance)¹², il est clair que les connaissances tacites conservent une place importante¹³, dont la sous-estimation est souvent la cause d'échecs lors des opérations d'automatisation ou de transferts de technologies (Donald Lambertson, 1997, p. 77) : "de nombreuses tentatives de remplacer des êtres humains (...) par des outils automatisés, échouèrent lorsqu'on s'aperçut plus tard que ce qui semblait être une tâche purement mécanique requérait au contraire, pour être accomplie de manière efficace, la capacité des humains d'agir au sens plein du terme" (Harry M. Collins, 1992, p. 156).

¹⁰ "Le rôle de nouvelles technologies de l'information ne doit pas être considéré comme un substitut des compétences humaines et des connaissances tacites. L'utilisation intelligente des technologies de l'information doit plutôt être envisagée comme une infrastructure de support à la formation et à l'utilisation des connaissances tacites" (Bengt-Ake Lundvall, 1997, p. 26-27).

¹¹ Notamment sous la forme du savoir tacite, que représentent les compétences nécessaires au traitement du savoir codifié (OCDE, 1996, p. 14).

¹² Si mesurer le stock de capital physique dont dispose l'économie est une tâche colossale, mesurer le stock du capital de connaissances paraît, *a fortiori*, une entreprise presque impossible (OCDE, 1996, p. 33). Il faut en effet prendre en compte l'accélération du taux de croissance des stocks de savoir accessible, mais également le renouvellement accru du stock de savoir, car les déperditions et l'obsolescence augmentent (idem, p. 13).

¹³ De même, Harry M. Collins souligne que "le rapport entre ce que nous pouvons dire sur ce que nous savons et ce que nous savons n'est pas un rapport statique" : s'il augmente avec "l'explication du savoir de l'homme de l'art grâce à la recherche scientifique (...), il diminue au fur et à mesure que ce qui était autrefois étrange et nouveau est incorporé dans le sens commun implicite, processus qui se produit aussi bien au niveau social qu'au niveau de l'intériorisation individuelle du savoir-faire" (1992, p. 148 et 154).

C'est pour cette raison que Giovanni Dosi et Luigi Orsenigo, (1988, p. 15) élargissent la définition traditionnelle de la technologie comme un ensemble d'informations relatives à certaines combinaisons d'inputs, en y intégrant les éléments d'un savoir difficilement codifiable sur l'utilisation de ces inputs et les processus de recherche et d'apprentissage. La sous-estimation du rôle des connaissances tacites a plusieurs explications. Premièrement, les connaissances tacites sont par définition plus difficilement identifiables et observables. Deuxièmement, les connaissances les plus facilement codifiables sont celles "autrefois considérées comme appartenant au domaine réservé d'une élite" (Harry Collins, 1992, p.153) à la différence de nombreux savoir-faire perceptifs et manuels (habileté technique), moins valorisés, et dont l'acquisition nécessite des contacts personnels et la participation à des activités suivies (Donald Lambertson, 1997, p. 78) : il est plus facile d'automatiser le calcul d'un logarithme que la reconnaissance d'une pièce défectueuse. Enfin, le maintien d'une part importante de connaissances tacites a également des explications socio-économiques. Par exemple, si pour une organisation, la codification de ses connaissances peut permettre une extension et une rationalisation de leur utilisation, elle peut également favoriser leur diffusion à l'extérieur de cette organisation, ce qui peut lui être préjudiciable dans les cas où cette connaissance constituait un actif stratégique, source d'un avantage concurrentiel (Jean-Claude Tarondeau, 1998). De même, il est nécessaire d'intégrer le caractère stratégique de la codification des connaissances : celle-ci nécessite le plus souvent la collaboration active des détenteurs des connaissances tacites ; or les détenteurs de la connaissance tacite avant codification ne sont pas nécessairement les mêmes acteurs que ceux qui détiendront la connaissance codifiée (Pascal Petit, 1998, p. 377).

B - L'INFORMATION, ELEMENT CENTRAL D' UN PROCESSUS DE COMMUNICATION DE CONNAISSANCES, DIRECT OU MEDIE

1 - L'information, une position intermédiaire entre données et connaissances ?

Une première possibilité de délimitation de l'information consiste à considérer que l'information décrit une réalité intermédiaire entre les données et les connaissances (au centre du schéma I). Par rapport aux données, l'information se caractérise par une mise en forme (étymologiquement information vient du latin *in forma*), dans des formats stabilisés et dans des environnements normalisés, des données. Les mêmes données peuvent permettre la production d'informations "de formes" différentes : par exemple, des données météorologiques peuvent être exprimées sous la forme d'un tableau statistique, d'un texte,

d'une communication orale, d'une carte, d'une animation. L'information est donc porteuse d'un contenu sémantique plus riche que les données qu'elle intègre et marquée par la subjectivité du producteur de l'information. Elle est une donnée qui acquiert du sens, un "signifié" porté par la donnée.

Par rapport aux connaissances, l'information peut être définie comme une formalisation de celles-ci : "une information est une connaissance mise en forme, c'est à dire inscrite dans un support, plus ou moins durable, qui permet la transmission" (Michel Callon, 1993, p.5)¹⁴. On retrouve les phénomènes d'objectivation et de réduction des connaissances, caractéristiques de la codification. A partir de là, "l'information peut se classer en deux groupes : dans le principal, l'information se compose de simples données, immédiatement utiles pour l'agent récepteur (information directe) ; dans le deuxième, l'information est plus subtile : elle décrit des connaissances, c'est à dire qu'elle représente des structures et/ou des procédures cognitives qui, une fois réalisées et mises en œuvre, génèrent enfin les éléments d'information utiles au récepteur" (Jean-Daniel Dessimoz, 1997, p. 224).

2 - Pour une définition de l'information comme l'élément central d'un processus de communication de connaissances ...

Si cette conception de l'information permet une première différenciation par rapport aux données et aux connaissances, elle doit être complétée par le fait que l'information est inséparable de la communication : "une information qui ne serait potentiellement communicable à aucun récepteur, fût-il l'émetteur lui-même, est proprement inconcevable" (Jean-Louis Le Moigne, 1998, p. 59). L'idée d'information implique la présence d'un émetteur, d'un (ou de plusieurs) récepteur(s) et l'existence d'un canal entre eux. On retrouve la vision de l'information comme une "séquence de signaux, correspondant à des règles de combinaisons précises, transmises entre un émetteur et un récepteur par l'intermédiaire d'un canal" (Dictionnaire Larrousse). L'information est intimement liée aux notions de flux, de processus (Bernard Lamizet, Ahmed Silem, 1997, p. 297) distinctes de la notion de stock, que celui-ci soit considéré sous un angle objectif (données) ou subjectif (connaissances). Ce flux de messages peut être très divers, depuis la simple transmission de données sans grand effort de production (de mise en forme) et sans grande mobilisation cognitive du récepteur

¹⁴ Conception partagée par Partha Dasgupta et Paul A. David (1994) pour qui "l'information est de la connaissance qu'on a mise sous la forme de messages qui peuvent être transmis à des agents", et par Jacques Arsac (1987) qui définit l'information comme "la forme qui porte la connaissance".

(information élémentaire) jusqu'à des informations complexes constituant un transfert de connaissances. C'est à cette conception de l'information que se réfère Fritz Machlup (1962) quand il énonce qu'il ne retient aucune distinction pertinente entre information et connaissance quant à leur objet, la seule différence se situant par rapport au temps, de façon analogue à la distinction entre flux et stocks d'un système dynamique et évolutif. C'est également à cette conception que renvoie l'affirmation de Kenneth E. Boulding (1988), selon laquelle "l'information est à la connaissance ce que le revenu est au capital" ; si on l'envisage du point de vue du récepteur de l'information, on peut ajouter que l'information est également à la connaissance ce que l'investissement est au capital. La théorie de la mesure probabiliste de l'information de Claude Shannon (1949) est également basée sur la vision de l'information comme un flux, mais en ne prenant pas en compte la signification de l'information, elle se limite à la mesure d'un flux de données, à l'étude des propriétés de la transmission de signaux, et à des questions de fiabilité du canal et de reconstitution du message entrant¹⁵.

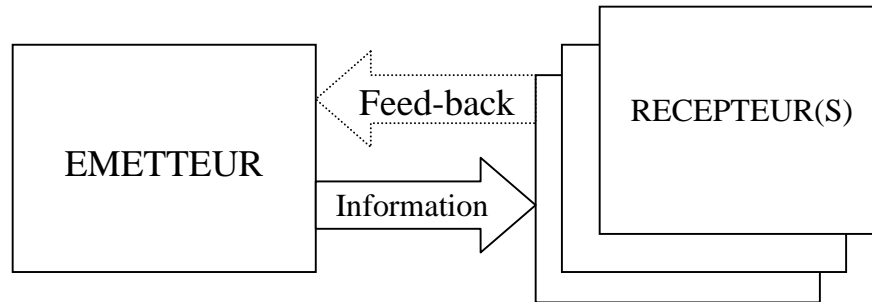
Or, ce qui fait que l'information est plus qu'un flux de données, c'est qu'elle est porteuse de sens¹⁶ : "un signe sans signification pour un récepteur ne peut être tenu pour une information" (Jean-Louis Le Moigne, 1998, p. 57). Ce sens - inséparable d'un contexte - que lui attribue l'émetteur (ou destinataire dans la terminologie de R. Jakobson) de l'information, et qui est interprété par le(s) récepteur(s) (ou destinataire) de l'information également situé(s) dans un contexte qui peut être différent de celui de l'émetteur, implique que le flux n'est pas une simple transmission technique entre individus passifs mais un processus de communication sociale entre acteurs situés dans un environnement (Pierre-Yves Gomez, 1997, p. 65). Nous définissons donc *l'information comme l'élément central d'un processus de communication de connaissances, processus qui peut se décomposer en un processus de production et un processus d'acquisition* (Pascal Petit, 1998, p. 28). Ces deux phases peuvent être inextricablement liées lors d'une communication directe, ou séparées dans le temps et/ou dans l'espace lors d'une communication médiée.

¹⁵ Du reste, l'objectif de Claude Shannon n'était pas d'élaborer une théorie de l'information mais une *théorie mathématique de la communication* (titre de son ouvrage fondamental). Ses travaux sur l'existence de codes économiques qui minimisent la quantité d'information à transmettre sur une ligne de communication, et sur les codes correcteurs d'erreur permettant de compenser le "bruit" sur une ligne de communication, ont joué un rôle majeur dans la conception des ordinateurs. Kenneth Boulding appelle l'information étudiée par Shannon une information "Bell Telephone", en comparant l'évaluation de Shannon à un relevé de téléphone, calculé sur la base du temps et de la distance mais qui ne dit rien du contenu de l'information.

¹⁶ Olivier Favereau (1998, p.196) va jusqu'à définir l'information comme "toute source de sens dans une théorie générale de l'activité humaine".

3 - ...qui peut être direct...

Schéma II : L'information comme élément central d'un processus de communication directe de connaissances



La première modalité correspond à une situation de face à face entre émetteur et récepteur (conversation, démonstration, conseil, face à face pédagogique). Ce mode de transmission direct des connaissances est présenté comme "le système le plus fondamental et le plus sophistiqué pour accomplir l'intelligibilité mutuelle, en exploitant les ressources linguistiques, ostensives et inférentielles" (Suchman¹⁷). Il comporte en effet plusieurs avantages : premièrement, par l'utilisation de canaux physiologiques différents (tactile, auditif, visuel), il peut permettre, par imitation et apprentissage, la transmission de connaissances tacites non exprimables hors de l'action de celui qui les détient. Lorsqu'il est principalement basé sur une information orale, les autres dimensions informatives de la communication ne doivent pas être sous-estimées ; cette importance est perçue à contrario lorsque ces autres formes d'informations sont absentes comme lors d'une communication téléphonique. Du reste, c'est l'efficacité permise par le recours simultané à des formes d'informations différentes qui explique le développement, dans les communications médiées, du multimédia¹⁸. Deuxièmement, par l'existence d'une interaction directe entre émetteur et récepteur(s), il autorise les effets de *feed-back* qui permettent à l'émetteur d'ajuster, en temps

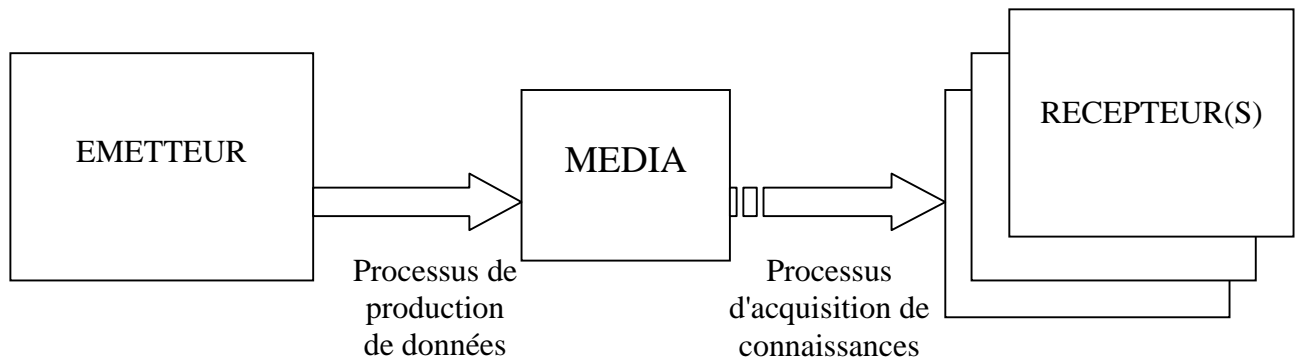
¹⁷ Cité par Bernard Conein, 1997, p. 36.

¹⁸ Fabio Arcangeli et Christian Genthon (1997) insistent sur les possibilités apportées par le multimédia pour communiquer des connaissances tacites en faisant l'économie du processus de formalisation de celles-ci ("flux de connaissance tacite sur les canaux multimédias", p. 165). Il nous semble toutefois que ce qui est inscrit sur un support, même multimédia, ce sont des connaissances codifiées : les possibilités qu'offre cette forme de codification facilitent les processus que Ikujiro Nonaka et Hirotaka Takeuchi (1997) nomment externalisation (conversion des connaissances tacites en connaissances codifiées) et internalisation (conversion des connaissances codifiées en connaissances tacites), elles ne les suppriment pas.

réel, les informations fournies aux réactions des récepteurs. Par contre, en ne mobilisant que des "informations vives" (Daniel Dufourt, 1997, p.157), ce processus de communication se limite aux récepteurs présents à l'endroit et au moment où s'effectue l'émission d'informations.

4 - ...ou médié

**Schéma III : l'information comme élément central
d'un processus de communication médiée de connaissances**



La deuxième modalité de communication, en introduisant un média entre l'émetteur et le(s) récepteur(s) de l'information, permet un décalage spatio-temporel entre un processus de production de données transmissibles et un processus d'acquisition de connaissances. Ce média, face matérielle du message, est un canal technique qui peut être sonore (téléphone, radio), visuel (photo, livre, journal, panneau, tableau), audiovisuel. Le processus de production de données aboutit à "une information morte dont le contenu défini une fois pour toutes, peut être transféré indépendamment de son concepteur" (Daniel Dufourt, 1997, p.157). Il correspond à la production de nouvelles données inscrites sur un support physique, de plus en plus facilement reproductibles à bas coût. Par contre à la formalisation des connaissances toujours nécessaire pour des raisons cognitives (faciliter la compréhension des récepteurs), s'ajoutent des impératifs de codification pour des raisons techniques liés aux caractéristiques du média utilisé. "L'émetteur doit procéder à une création d'un message, extrait, déconnecté de sa base de connaissances, mis en forme de façon appropriée au média utilisé pour la communication, et adapté aux capacités d'assimilation qu'il prête aux destinataires potentiels de l'information ; l'émetteur doit donc se faire une représentation du processus de communication à venir, entachée d'incertitude" (Marie de Besses, 1999, p.262). La plus grande efficacité en terme d'extension du nombre de récepteurs potentiels doit être comparée

à une moindre efficacité en termes de dégradation possible du contenu sémantique des connaissances communiquées.

Cette efficacité repose sur des avantages différents dans les deux situations (possibilité d'une interaction directe dans une situation de face à face, existence d'un support durable lors d'une communication médiée) et de nombreuses situations concrètes de communication de connaissances combinent les deux modalités (enseignement avec support de cours, colloque avec distribution des communications, vente de brevets avec services d'aides au transfert de technologies...).

C - L'EFFICACITE DU PROCESSUS INFORMATIONNEL

L'analyse des "bruits sémantiques" dans les processus informatifs, c'est à dire le fait "qu'un message peut être reçu sans que la signification décodée soit celle programmée à l'émission" (Bernard Lamizet, Ahmed Silem, 1997, p. 121), conduit aux constatations suivantes.

Premièrement, le récepteur de l'information a un rôle déterminant dans l'efficacité du processus, et en ce sens on peut considérer qu'il est coproducteur de l'information (Anne Mayère, 1990, p.56)¹⁹. "Ce qu'on appelle de façon simplifiée mémorisation de l'information par un sujet est une appropriation²⁰, une incorporation à ses structures cognitives (...) [qui] dépend des conditions d'acquisition de l'information (par exemple, opinion du sujet sur la fiabilité du canal), des capacités de traitement de l'information par le sujet (de ses structures cognitives présentes sur le domaine), et, plus largement, du contexte social dans lequel il peut mobiliser cette information" (Marie de Besses, 1999, p.262). En conséquence, "des agents différents peuvent déduire des connaissances différentes, voire même contradictoires, d'un même ensemble d'informations" (Patrick Cohendet, 1997, p. 104). Interviennent les compétences distinctes des utilisateurs et le contexte différencié de la réception de l'information : il s'agit d'une cognition située (Bernard Conein, Laurent Thévenot, 1997). La

¹⁹ "La qualité de l'information est tellement liée aux caractéristiques de son utilisateur potentiel que ce dernier devra nécessairement être impliqué en profondeur dans son élaboration" (Olivier Favereau, 1998, p. 204).

²⁰ Armand Hatchuel (1996) opère une distinction entre "l'acquisition de la connaissance" qui existe dans tout processus informatif et "l'appropriation de la connaissance" qui, seule, est créatrice de richesse.

relativité contextuelle du sens d'une information est illustrée par Umberto Eco en prenant l'exemple d'une même conférence sur l'institution monarchique : celle-ci change de signification en changeant de lieu, selon que les mêmes mots sont prononcés à Stockholm, à Rabat ou à Paris (Olivier Favereau, 1998, p. 220). Les compétences du récepteur incluent des compétences techniques concernant la maîtrise de l'utilisation du média, mais, surtout un ensemble de savoirs très divers, indispensables pour filtrer, restructurer, interpréter, mettre en rapport l'information transmise et enrichir son stock de connaissances (Kenneth E. Boulding, 1988). C'est pourquoi la connaissance que s'approprie le récepteur peut ne représenter qu'une part limitée de la connaissance émise, ou qu'à l'inverse, dans la situation particulière d'un récepteur beaucoup plus compétent que l'émetteur, la connaissance appropriée peut être plus importante que la connaissance émise²¹. La dégradation sémantique de la connaissance transmise peut être particulièrement élevée quand il existe plusieurs intermédiaires successifs entre l'émetteur initial et le destinataire final. Le rôle décisif du récepteur a été abondamment analysé dans le cas plus général d'un processus de communication quelconque : c'est en étudiant la diversité de perception d'un même message que les chercheurs en communication de Palo Alto ont démenti les prévisions de standardisation des individus qui devait résulter inévitablement du développement des médias de masse selon les théoriciens de l'Ecole de Francfort.

Deuxièmement, l'importance respective de l'émetteur et du récepteur est très variable selon les processus concrets d'informations. A la situation extrême des médias de masse où le récepteur est essentiellement passif, peut être opposé le rôle actif du chercheur, qui, en recherchant des données, initie le processus et produit un "effet de connaissance" (Jean-Claude Passeron, 1991). Cette diversité se retrouve dans les différents usages possibles de la mise en réseau d'une masse considérable de données, permise par le développement d'Internet : recherche de données pertinentes par un utilisateur souhaitant s'informer (*self média*) d'une part, informations "poussées" vers des utilisateurs dont on a au préalable identifié le profil (technologies *push*) d'autre part. Le niveau actuellement relativement élevé de compétences des utilisateurs d'Internet peut expliquer pourquoi, pour le moment, c'est le premier modèle qui domine.

²¹ L'idée de connaissance que s'approprie le récepteur du processus informatif ne doit pas induire une vision d'un accroissement linéaire de son stock de connaissances. Dans certains cas, la connaissance acquise invalide une partie des connaissances que détenait le récepteur.

Troisièmement, des connaissances communes²² à l'émetteur et au récepteur de l'information nécessitent un flux moindre de données pour communiquer une certaine quantité de connaissances. L'existence d'une certaine connaissance commune, notamment langagière, est indispensable pour que puisse exister un processus de communication. Mais quand cette connaissance est limitée, la masse de données à transmettre (en quantité et en variété) doit être abondante pour assurer une communication efficace, ce qui peut nécessiter des moyens techniques importants. A l'inverse, des connaissances partagées entre l'émetteur et le récepteur, qui peuvent rester implicites, permettront de minimiser les données transmises. Le contexte de la communication joue un rôle important pour déterminer ce qui est commun : Nicholas Negroponte (1995, p 48) donne l'exemple d'un clin d'œil échangé entre un mari et sa femme au cours d'un dîner (un bit de donnée) qui peut correspondre à 100 000 bits de données s'ils avaient dû être explicités. De même, une prise de notes visant à se transmettre des connaissances dans le temps, nécessite d'inscrire une quantité de données minimales dans la mesure où l'émetteur et le récepteur de l'information sont la même personne.

Quatrièmement, au-delà des connaissances communes, c'est l'intercompréhension existante entre l'émetteur et le récepteur qui est le facteur déterminant de l'efficacité du processus informatif, la connaissance qu'a un acteur du processus de ce que sait (et ce que ne sait pas) l'autre acteur, la capacité à "se mettre à sa place" : "l'information n'est reliée ni à des collectifs objectivants, ni à des individus connaissant, mais à une coordination problématique" (Laurent Thévenot, 1997, p. 208). L'émetteur comme le récepteur doivent consacrer au moins autant d'effort et d'intelligence à l'analyse des conditions sociales d'émission, de communication, d'acquisition et de mobilisation d'un message, qu'au contenu de celui-ci.

Pour dépasser ces quelques remarques et analyser véritablement l'efficacité du processus informationnel, il faudrait préciser les cadres institutionnels permettant de produire ces connaissances partagées et cette intercompréhension entre émetteur et récepteur(s) de l'information, ce qui, dans le cas de l'information en général, dépasserait le cadre de cette étude. Avant d'examiner les particularités du traitement automatique de l'information, il est

²² La notion de connaissances communes est considérée ici dans son acceptation courante et non dans le sens qui lui est attribué par la théorie des jeux, où tout le monde possède la connaissance en question et le sait. Les chercheurs en communication emploient également l'expression de répertoires communs.

nécessaire d'évoquer quelques conséquences sur l'économie de l'information, que nous utiliserons par la suite sur le cas particulier des logiciels.

D - QUELQUES CONSEQUENCES SUR L'ECONOMIE DE L'INFORMATION

La définition de l'information comme élément central d'un processus de communication de connaissances (direct ou médié) peut permettre d'apporter quelques éléments de clarification aux questions complexes de l'analyse économique de l'information : "située aux frontières de l'économie, au carrefour des biens matériels et immatériels, des produits marchands et non marchands, objet consommé mais non détruit par la consommation, coûteuse à constituer mais facile à dupliquer (...), l'information se laisse difficilement saisir (...) et accumule devant l'économiste les motifs de découragement" (Commissariat Général du Plan, 1990).

Pour beaucoup d'économistes, l'information posséderait des caractéristiques intrinsèques tellement différentes des autres biens et services qu'il serait impossible de lui appliquer les catégories habituelles de l'analyse économique (produit, valeur, prix, marché, concurrence). Sans nier certaines spécificités de l'information, on peut se demander s'il n'est pas plus pertinent de modifier le cadre conceptuel dominant, centré sur des produits industriels, pour pouvoir appréhender toute la diversité des biens et des services, incluant l'information²³.

Sont par exemple souvent mis en avant concernant l'information, les difficultés, voire l'impossibilité, pour définir des unités de produit, pour connaître la qualité du produit avant de l'avoir consommé, pour séparer les phases de production et de consommation, pour identifier les apports respectifs des coproducteurs, pour attribuer une valeur à un produit dont la valeur d'usage est très différente selon l'utilisateur²⁴. Or ces incertitudes majeures se retrouvent dans de nombreuses situations de services, notamment celles où la *relation de service* est importante (Jacques De Bandt, Jean Gadrey, 1994), voire dans certains cas pour l'ensemble des biens. Par exemple, si la valeur d'usage d'une information est effectivement très

²³ Ainsi Anne Mayère, après avoir observé que l'information échappe au statut "classique" de la marchandise, se demande si ce n'est pas plutôt ce statut classique qu'il faut remettre en cause : le "bien", opposé aux spécificités de l'information, est-il "représentatif de ce que sont les marchandises qui s'échangent ?" (1997, p. 125).

²⁴ Voir le *survey*, effectué par Sandra Braman, des problèmes rencontrés lorsque l'on mobilise un cadre d'analyse néoclassique (1997, p. 89-102).

dépendante de la personnalité de son récepteur qui est ou non capable d'en exploiter les potentialités, cette subjectivité existe également pour une part importante des biens, la variété des besoins et des moyens de les satisfaire n'étant guère moins importante que celle des compétences (Jean-Daniel Dessimoz, 1997, p. 221). De même, ce n'est qu'*ex post* que l'on peut apprécier la qualité de nombreux services comme un repas au restaurant ou un circuit touristique, activités pour lesquelles les phases de production et de consommation ne peuvent être disjointes. Par contre, l'affirmation de Kenneth J. Arrow (1962) selon laquelle "l'acheteur ne connaît la valeur de l'information que lorsqu'il se l'est approprié"²⁵ semble quelque peu exagérée : dans de nombreux cas on peut décrire des caractéristiques pertinentes (objet, méthode, source) d'une information sans la dévoiler. Concernant ce problème (de détermination *ex ante* de la valeur d'une information), Joseph Stiglitz (1985) considère qu'il s'agit d'une situation de régression infinie, dans la mesure où il est impossible de déterminer s'il vaut la peine d'obtenir des informations sur l'intérêt qu'il y a à obtenir des informations... Il nous semble que ce raisonnement constitue davantage une mise en évidence des limites d'une vision substantielle de la rationalité, concernant la consommation de l'ensemble des biens et services, qu'une spécificité des produits informationnels.

Par contre, certaines spécificités économiques de l'information peuvent être analysées à partir de la distinction entre processus de communication directe et processus de communication médiée.

Dans le cas où la relation est directe entre l'émetteur et le(s) récepteur(s) de l'information, celle-ci peut être assimilée sur le plan économique à la fourniture d'un service. L'argument selon lequel l'information aurait un caractère inappropriable, dans la mesure où toute personne qui dispose d'une information ne la perd pas en la transmettant²⁶, semble biaisé par une confusion entre connaissance et information. En effet, l'objet de l'échange est l'information, c'est à dire le processus de communication des connaissances, et non les connaissances de l'émetteur, qui effectivement ne diminuent pas (elles peuvent même augmenter en conséquence des efforts de formalisation et des effets de *feed-back* éventuels

²⁵ Le raisonnement de Kenneth Arrow est le suivant : la valeur de l'information reste pour l'acquéreur inconnue jusqu'à ce qu'il l'obtienne, mais au moment où il l'obtient, il a en fait acquis gratuitement la connaissance elle-même.

²⁶ C'est le type d'argument qu'utilise René Mayer (1997) quand il dit que la notion de consommation de l'information est problématique en raison de la non-destruction de l'information consommée et du maintien de la possession de l'information communiquée (p. 62-63).

lors de ce processus). Cette distinction n'est guère différente de celle que l'on peut opérer dans le cas d'un coiffeur entre une coupe de cheveux et les compétences du coiffeur. On peut juste noter que dans le cas de l'information, dans la mesure où la prestation consiste à communiquer une partie des connaissances de l'émetteur, la valeur d'échange de celles-ci peut être modifiée : dans les situations où cette valeur était basée sur des compétences exclusives ou peu répandues, elle peut diminuer ; par contre, dans d'autres situations, le fait que des connaissances soient possédées par davantage de personnes peut augmenter leur valeur en élargissant leur espace de validité (externalité de réseau). Du reste, le fait que dans certaines circonstances l'information est fournie majoritairement de façon non marchande (l'enseignement), alors que dans d'autres elle est principalement marchande (le conseil), n'est pas dû à des différenciations techniques mais est le produit de conventions sociopolitiques.

Paradoxalement, l'analyse économique est plus complexe dans les cas où les processus de production et d'acquisition sont séparés. L'existence d'un média peut permettre l'inscription des données produites sur un support qui peut circuler indépendamment de l'émetteur. Le processus informatif peut donc se scinder en des processus économiques distincts effectués par des acteurs économiques différents : producteurs de données (appelés également producteurs ou éditeurs de "contenus"), distributeurs, opérateurs de transports de données, entreprises fournissant des services divers d'aide à l'acquisition (recherche, utilisation). De surcroît, "l'original" des données constitue un bien intangible (Peter Hill, 1997), qui possède deux des caractéristiques des biens publics (indivisibilité et bien non rival)²⁷, la troisième caractéristique (non-exclusion de l'usage) étant satisfaite de façon différenciée selon leur mode de diffusion (livres, cassettes, ondes hertziennes, réseaux informatiques). Si dans certains cas l'évolution technique a permis d'individualiser leur consommation (décodeurs de télévision, télévision par câble), dans d'autres cas les dispositifs de protection se sont avérés contradictoires avec les exigences d'efficacité d'utilisation (logiciels pour micro-ordinateurs). Les possibilités qu'offrent la numérisation de l'ensemble des données et l'extension des réseaux informatiques (copie et diffusion à un coût qui tend vers zéro, facilité d'intégration de données de source et de nature diverses) rendent le contrôle du droit de propriété principalement dépendant de mécanismes juridiques, divers selon les pays et la nature des

²⁷ L'indivisibilité renvoie au fait que les dépenses pour produire des données sont indépendantes du nombre d'utilisateurs de ces données. Une donnée est un bien non rival dans la mesure où la consommation de cette donnée par un utilisateur ne détruit pas celle-ci, ou dit autrement, la consommation par un utilisateur n'empêche pas sa consommation par d'autres consommateurs.

données (droit d'auteur, copyright, brevets) et complexes à mettre en œuvre (Jean-Daniel Dessimoz, 1997, p. 218). Blaise Cronin (1997) souligne les problèmes de tarification qu'induit "la facilité avec laquelle les biens et services informationnels se prêtent à la duplication, à la recombinaison, à la contrefaçon et à une large diffusion" (p.15).

A côté de ces considérations technico-juridiques, il faut également prendre en compte des traditions historiques, qu'illustrent par exemple les difficultés qu'ont eues les entreprises qui voulaient assurer par un financement direct les services de recherche de données sur Internet. De ce fait, les modèles économiques de l'information médiée sont très divers, hétérogènes, instables et en pleine évolution : à côté des relations marchandes "classiques", existent le financement public, le financement indirect (principalement par la publicité), les stratégies de *bundling* consistant à associer à la fourniture de données des services liés de support et d'usage²⁸, les stratégies de mise à disposition en temps réel de connaissances dont la valeur se déprécie très rapidement, l'émergence de modèles de don contre don avec les possibilités de réversibilité des rôles d'émetteur et de récepteur d'informations. Le foisonnement d'initiatives non marchandes, notamment dans les nouveaux processus informatifs permis par le développement d'Internet, ne doit toutefois pas masquer le fait que dans certains cas, c'est la marchandisation de l'information qui s'étend en raison de certaines évolutions techniques (les possibilités de micropaiement) ou sociopolitiques : privatisation de nombreux types d'informations auparavant publics comme l'accès aux bases de données gouvernementales aux Etats-Unis (Sandra Braman, 1997, p.105) ou aux données météorologiques en France. La distinction opérée entre données et informations met en évidence la divergence croissante entre l'augmentation illimitée du stock de données et de connaissances codifiées accessibles, et les capacités nécessairement limitées d'absorption d'informations des acteurs économiques ; de ce fait, dans les processus informatifs médiés, les sociétés proposant des services de recherche et de sélection adaptés aux besoins des utilisateurs acquièrent un poids économique de plus en plus important, ce qu'illustre la place prise par les "sites portails" et les moteurs de recherche dans Internet.

L'analyse concrète de ces différents modèles ne peut être effectuée qu'au niveau de chaque champ de l'économie de l'information. C'est ce que nous effectuerons dans le cas de

²⁸ Ces stratégies peuvent être analysées comme la tentative de réunir dans un même espace économique l'ensemble du processus informatif (de production, de transmission et d'acquisition).

l'économie du logiciel. Mais auparavant, il est nécessaire de définir le rôle des logiciels à partir des particularités du traitement automatique de l'information.

Section II - Les particularités du traitement automatique de l'information

L'informatique a tout d'abord augmenté considérablement les possibilités de stockage et de transmission des données. Mais, de ce point de vue, l'ordinateur²⁹ ne joue pas un rôle qualitativement différent des autres médias, si ce n'est que la croissance exponentielle des capacités de mémorisation et de transmission, et la baisse de leurs coûts, ont considérablement amplifié la portée des questions analysées précédemment.

L'originalité fondamentale des systèmes informatiques est qu'à la différence des autres médias, où s'opèrent simplement des opérations de codage et de décodage (ou de modulation et de démodulation) des données pour faciliter leur transmission mais avec restitution de données identiques, il existe la possibilité d'effectuer lors du processus informatif un traitement automatique des données et donc une modification de celles-ci.

En conséquence l'information émise en direction d'un système informatique est particulière. Certes comme toute émission d'informations elle vise à déclencher une action du récepteur, caractérisé comme étant un décideur³⁰, ce qui constitue la composante pragmatique de l'information inséparable de ses composantes syntaxiques et sémantiques (Jean-Louis Le Moigne, 1998, p.56). La définition de l'information de Partha Dasgupta et Paul David (1994) comme étant de "la connaissance mise sous la forme de messages" précise que ces messages "déclenchent [chez les récepteurs] des décisions qu'ils n'auraient pas prises ou qui auraient eu un contenu différent s'ils n'avaient pas reçu ce message".

²⁹ L'ordinateur est né de la rencontre entre la codification de l'information et l'utilisation de l'électricité comme support informationnel (électronique).

³⁰ "L'information peut donc être définie comme tout stimulus modifiant les connaissances du décideur" (David B. Lawrence, 1997, p 200).

Mais quand le récepteur de l'information est un acteur humain, il interprète cette information, il lui attribue une signification et en conséquence son action n'est pas complètement prédéterminée : "l'une des caractéristiques de l'information, c'est de pouvoir *a priori* être interprétée librement, de pouvoir donner lieu à plusieurs interprétations, de déboucher sur plusieurs types de réactions ou de décisions" (Bernard Paulré, 1996, p. 12).

Par contre, quand ce récepteur est un ordinateur, il n'affecte aucun sens aux données qu'il reçoit : "Un ordinateur est une machine qui traite des impulsions électroniques. Ces impulsions représentent des données qui n'ont de sens que pour l'homme. Cette machine est "stupide", ne connaît pas la fantaisie, et n'interprète jamais les instructions qu'on lui donne. Son unique finalité est l'exécution entièrement déterministe des programmes qu'on lui confie" (Philippe Breton, 1990, p. 58). Ce que traite l'ordinateur ce sont des *données*, indépendamment de la signification qu'un récepteur humain pourrait leur attribuer³¹, en suivant les instructions de logiciels écrits *préalablement* à l'émission d'informations de l'utilisateur, et donc en anticipant le contenu de ces entrées de données.

Le fait que, dans un processus informatif reposant sur un système informatique, les données subissent des modifications a pour conséquence que dans de nombreux cas l'émetteur et le récepteur final sont la même personne. Dans un premier temps nous analyserons cette situation, plus simple à représenter, tout en soulignant que dans les cas où l'émetteur et le récepteur sont des personnes différentes, l'efficacité du processus informatif devra également intégrer la qualité de leur intercompréhension analysée précédemment.

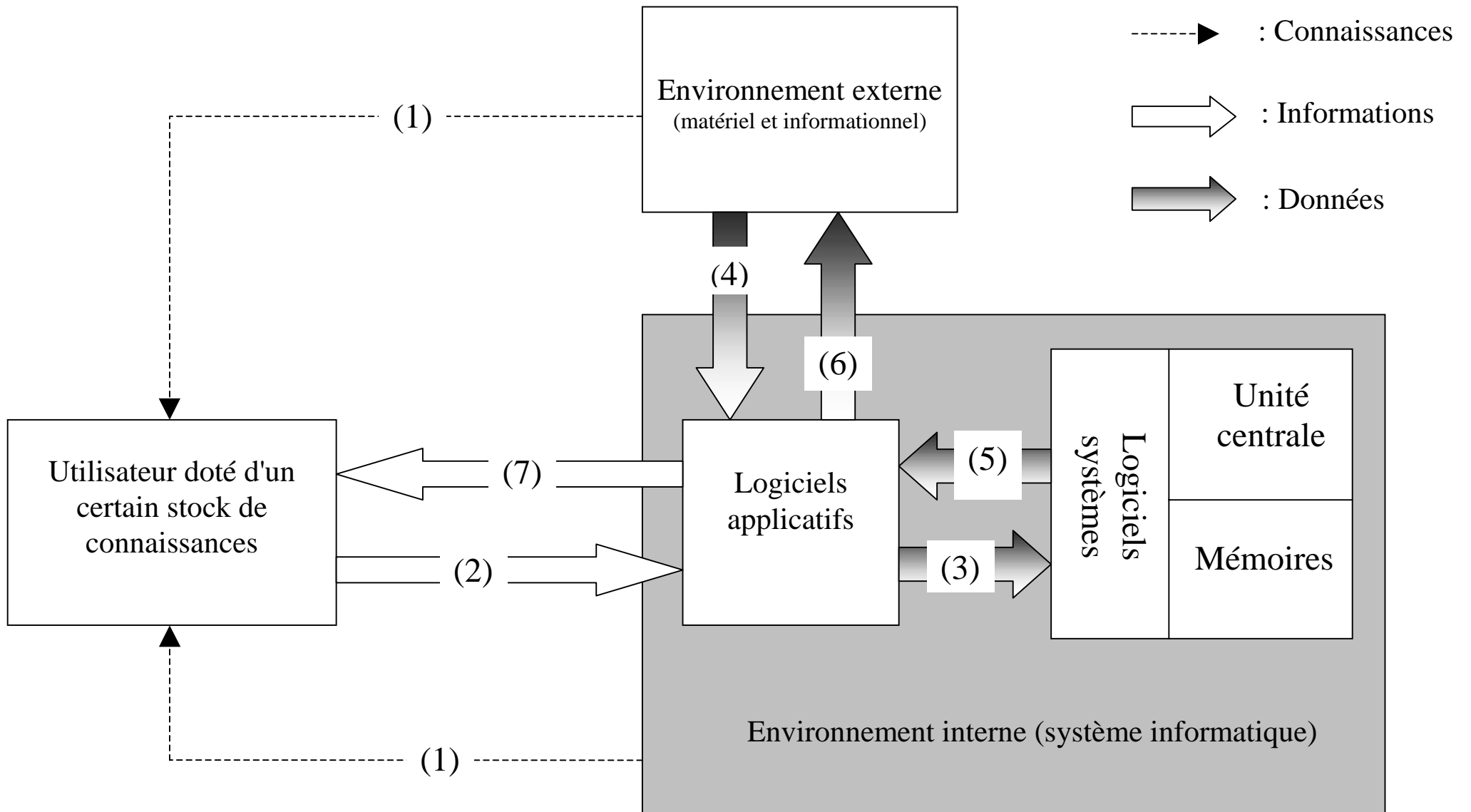
Avec les premiers ordinateurs, l'utilisateur, nécessairement informaticien, se sert du langage machine pour pouvoir traduire les opérations qu'il souhaite en "instructions" exécutées par l'ordinateur, introduit les données sous forme binaire (directement ou par l'intermédiaire de perforateurs) et récupère les résultats sous la forme également de données binaires (cartes perforées). Par exemple, sur le rudimentaire Altaïr, premier micro-ordinateur à avoir connu un certain succès commercial, les informations étaient rentrées en manipulant des commutateurs sur un panneau à l'avant du boîtier qui répondait par un clignotement d'une rangée de voyants. Progressivement, les intermédiaires vont se multiplier entre l'utilisateur et

³¹ On peut illustrer la différence entre la réception d'un flux de données par un être humain et par un système informatique en examinant les conséquences de la modification de quelques bits ; il est vraisemblable que dans le premier cas cela ne modifiera guère la signification attribuée au message par le récepteur humain, alors qu'une telle modification risque de perturber profondément le comportement de l'ordinateur.

l'ordinateur, permettant son utilisation par un public de plus en plus large : création de langage d'assemblage, puis de langage de programmation de *haut niveau* permettant d'écrire du code-source qui est converti en langage machine (code objet) par des assembleurs, interpréteurs ou compilateurs, création de systèmes d'exploitation et surtout d'applications très diverses répondant aux différents besoins des utilisateurs, création d'interfaces de plus en plus diversifiées entre l'homme et le système informatique.

Le traitement automatique de l'information est représenté par le schéma IV. Le traitement automatique de l'information est réalisé par un système, dans le sens où il est effectué par des produits complémentaires dont l'utilité provient de leur association en un ensemble cohérent. Un système informatique comporte des composants matériels (unité centrale, mémoire, périphériques, dispositifs d'interfaces) et des composants logiciels (logiciels systèmes et logiciels applicatifs) divers. Comme tout système finalisé, il est connecté à l'environnement extérieur par deux types de canaux : les canaux afférents, ou sensoriels (représentés par les flux 2 et 4 sur le schéma) par lesquels le système reçoit une information sur son environnement, et les canaux efférents, ou moteurs (représentés par les flux 6 et 7 sur le schéma), par lesquels le système agit sur son environnement (Herbert Simon, 1974, p. 85).

Schéma IV : Le traitement automatique de l'information



En commentant les différents flux du schéma, le traitement automatique de l'information, dont les primitives sont l'acquisition, la mémorisation, le traitement et la présentation (Monique Picavet, 1997, p. 30), peut être décomposé de la façon suivante:

(1) L'utilisateur d'un système informatique doit posséder une certaine compétence ; sont importantes ses connaissances du système informatique, acquises notamment par expérience, mais aussi ses connaissances du domaine considéré. L'ensemble constitue le cadre cognitif de l'utilisateur qui a une influence sur la pertinence des informations entrées et sur l'apport pour l'utilisateur des informations fournies par le système informatique.

(2) L'utilisateur émet des informations par l'intermédiaire de périphériques divers (clavier, souris, écran tactile, scanner, micro...) : ces informations émises se composent de données sous une forme adaptée aux possibilités du système informatique, et d'opérations à exécuter. L'information émise "devient donc donnée dès qu'elle est traitée, transformée ou mémorisée" (Monique Picavet, 1997, p. 56).

(3) Par des dispositifs d'interfaces pilotés par des logiciels, les données sont converties en données binaires pour pouvoir être traitées et les opérations sont traduites en instructions en langage machine ; l'ensemble de ces données et instructions élémentaires est stocké ensemble dans la mémoire vive de l'ordinateur (c'est une des originalités des principes architecturaux du modèle de Von Neumann sur lesquels sont construits tous les ordinateurs, et c'est ce qui permet d'adapter automatiquement le comportement d'un même programme en fonction des données introduites, et même de faire fabriquer un programme par un autre programme, comme dans le cas d'un compilateur). On peut considérer que ces données sont des *données vives* pour désigner le fait qu'elles sont créées lors du processus informatif par opposition aux *données mortes* créées et stockées antérieurement à ce processus.

(4) Pour réaliser automatiquement les opérations demandées, il peut être nécessaire d'utiliser d'autres données (mortes). Ces données peuvent être des données déjà stockées dans le système informatique (fichier sur un disque dur par exemple), être extraites de bases de données extérieures auxquelles l'ordinateur est connecté par un réseau, ou être des signaux transmis par des capteurs et mesurant certains états de l'environnement externe.

(5) Les données binaires qui résultent du traitement sont transmises à l'utilisateur par l'intermédiaire de périphériques de sortie (écran, imprimante, haut-parleur...) grâce à des dispositifs d'interfaces pilotés par les logiciels.

(6) Ces données peuvent être également stockées dans les mémoires de l'ordinateur (sauvegarde), ou alimenter des bases de données externes ou déclencher des actions physiques sur l'environnement par l'intermédiaire d'actionneurs (ou automates).

(7) En donnant une signification aux données qu'il perçoit, l'utilisateur reçoit une information qui enrichit son stock de connaissances et peut lui permettre d'agir sur l'environnement (aide à la décision) ou de réémettre des informations vers le système informatique, en poursuivant le processus informatif.

On peut faire ici deux remarques complémentaires. Premièrement, la diversité des usages de l'informatique fait que selon les cas, certaines de ces phases n'existent pas ou ont une importance très variable (le pilotage d'une centrale nucléaire est assez éloigné de l'utilisation d'un tableur). Deuxièmement, aux débuts de l'informatique, chacun des processus informatifs couvrant les différentes phases était nettement disjoint (traitement par lots ou mode différé). Avec le développement du traitement interactif (ou mode conversationnel), les différents processus deviennent beaucoup plus courts et s'enchevêtrent, facilitant l'utilisation des systèmes informatiques par l'émission permanente d'informations limitées en réaction immédiate aux informations reçues.

La conséquence la plus importante de l'existence d'un traitement des données dans le cas de l'informatique, par opposition à une simple transmission, est qu'à l'intercompréhension entre émetteur et récepteur d'information, condition d'efficacité de tout processus informatif, se superpose une intercompréhension nécessaire entre concepteurs du système informatique (notamment des logiciels) et utilisateurs (qu'ils soient émetteurs, récepteurs ou les deux à la fois). C'est ce que nous analyserons pour la production des logiciels. Auparavant il est nécessaire de préciser ce qu'est un logiciel.

Section III - Définitions et statuts des logiciels

Les logiciels jouent un rôle décisif dans ce processus. En pilotant des composants matériels divers (processeurs, mémoires, périphériques d'entrée et de sortie), les logiciels assurent la conversion des données entrées par l'utilisateur ou captées sur l'environnement en données binaires, proposent aux utilisateurs des actions à entreprendre, transforment les actions choisies en instructions exécutables par le processeur en combinant les données

entrées avec des données préalablement stockées, mémorisent les résultats de ces actions, les restituent aux utilisateurs sous une forme appropriée et commandent les actions directes sur l'environnement. Au-delà de ce rôle technique, les logiciels peuvent être appréhendés de différentes façons.

A - UN LOGICIEL EST UN TEXTE NUMERIQUE ACTIF

Indépendamment de leur diversité fonctionnelle, les logiciels ont en commun d'être des *textes numériques actifs*. Ecrits dans un langage de programmation, leur code source se présente sous la forme d'un *texte*³². La programmation est une forme d'écriture (Pierre Lévy, 1992, p. 10) et comme toute activité d'écriture, elle se caractérise par une diversité illimitée. Existant sous une forme *numérique*, le texte du programme possède de très grandes possibilités de modification, de réutilisation, de combinaison (Dominique Cotte, 1999). Mais ce texte est également un *texte actif*, un texte qui agit (Michel Callon, 1991, p. 205) dans la mesure où il se compose d'un ensemble d'instructions qui seront exécutées automatiquement par une machine. Le passage de l'univers humain à l'univers machinique s'opère dans la transformation (réalisée automatiquement par un compilateur ou un interpréteur) de son *code-source* (texte du programme écrit dans un langage de programmation compréhensible par l'être humain) en un *code-objet* (suite d'instructions en langage machine). Par cette opération, le logiciel se métamorphose d'une information particulière à destination d'une machine mais ayant un sens pour son concepteur, en de simples données numériques dépourvues de signification apparente, simple enchaînement d'actions effectuées "machinalement". D'une information sur la conduite d'un processus, d'une action, le logiciel devient lui-même processus, action (Jean-Benoît Zimmermann, 1995 B, p. 182). La forme la plus extrême de cette transformation est la gravure dans le silicium du code objet sous forme de composant électronique (bien tangible). Il est important de noter qu'à partir de l'unique disposition du code objet, il est très difficile de reconstituer le code source du logiciel par *reverse engineering* (décompilation) ; par contre la disposition du code source autorise une infinité de modifications permettant de générer facilement un programme au comportement différent.

³² "D'un point de vue syntaxique, un programme est un texte construit selon un ensemble bien défini de règles grammaticales. D'un point de vue sémantique, un programme exprime un calcul, ou pour être plus exact, un ensemble de calculs (...). Programmer consiste à transformer des spécifications qui décrivent une fonction en un programme, i.e., un texte qui peut être interprété par une machine afin de calculer cette fonction" (Françoise Détienne, 1998, p. 29).

B - UNE CREATION HUMAINE QUI DEVIENT UN "ACTEUR NON HUMAIN" ?

Un logiciel est bien évidemment une création humaine : "un programme a le même statut théorique qu'une petite théorie qui modéliserait un fragment de la réalité" (Jacques Printz, 1998, p. 46). Selon cet auteur, un programme "est le fruit de la pensée de son créateur et rien que cela : c'est de la sémantique à l'état brut" (idem, p. 234). Jacques Printz en déduit que "l'homme est indispensable au procédé de fabrication du logiciel" et, surtout, que "c'est lui seul qui fixe le sens des constructions qu'il élabore" (idem, p. 235).

Mais quand on est confronté à l'action d'un logiciel (ou plus généralement d'un système informatique), on peut considérer qu'on a affaire à des "*acteurs non humains*" (Michel Callon) avec lesquels l'utilisateur va devoir composer, négocier, ruser, manipuler. Cette conception d'un logiciel qui semble agir en s'étant autonomisé par rapport à son concepteur, est présente dans beaucoup d'études de l'informatique. Ainsi, Pierre Lévy considère qu'en informatique un acteur peut être un être non humain et définit le logiciel comme un acteur collectif (1992, p. 34). Philippe Breton souligne que "une grande partie de la magie de l'informatique tient dans ce déplacement simple qui s'opère avec constance : l'oubli qu'il y a là pure création humaine et la croyance que la machine est une sorte d'être par lui-même, qui aurait échappé à son concepteur" (1990, p. 67). Ce passage à une existence apparemment indépendante de leurs créateurs est spécialement net dans le cas des logiciels : "quand on écoute leurs créateurs, les programmes ont des intentions, font de leurs mieux, sont plus ou moins intelligents, ou stupides, communiquent entre eux et perdent le Nord" (Sherry Turkle, 1986). Tout en considérant que les "« choses » ne sont pas des « acteurs » en elles-mêmes", Jacques Girin (1994, p. 15) souligne que "les « choses » sont certainement des déterminants de l'action, tout autant que de simples outils pour l'action" ; les objets sont "non seulement des instruments" mais sont également "chargés de savoirs, de plans d'actions, de scénarios, etc. qui viennent à l'appui de ceux inscrits dans les cervelles humaines" (idem). Cette caractérisation du statut des objets nous semble particulièrement adéquate pour les logiciels.

C - LE LOGICIEL COMME "OBJET FRONTIERE"

Un logiciel est également un *intermédiaire* entre l'utilisateur et l'ordinateur qui permet de traduire les problèmes des acteurs humains dans le répertoire de la machine. "Un programme d'ordinateur est un message de l'homme à la machine. La syntaxe rigide et les définitions méticuleuses n'existent que pour rendre l'intention de l'homme claire pour la

machine stupide" (Frederick P. Brooks, 1996, p. 142). En tant que "forme d'expression humaine à destination d'une machine" (Gérard Dréan, 1996), le logiciel est un *objet frontière* (Geoffrey Bowker, Susan Leigh Star, 1997, p. 295) qui habite simultanément plusieurs mondes sociaux, depuis le monde de la création culturelle caractérisé par ses possibilités infinies, jusqu'au monde des produits et processus industriels aux contraintes techniques fortes. Les spécificités très différentes de ces mondes en termes de processus de production, de diffusion, d'utilisation, de droits de propriété sont une source permanente de tension dont il faut tenir compte. Le logiciel se situe à la frontière entre le "monde chaud" de la créativité humaine et le "monde froid" des objets techniques que sont les ordinateurs.

Un logiciel peut également être appréhendé comme une forme de résolution de problème, élaborée à partir d'une modélisation des connaissances formalisées et tacites que l'on possède sur un environnement. En ce sens un logiciel est un dispositif artificiel qui, en lien avec d'autres dispositifs matériels et logiciels, améliore les performances de la pensée humaine, un *artefact cognitif* (D. A. Norman, 1993) ou *système cognitif artificiel* (Jean-Daniel Dessimoz, 1997). En prolongeant l'analyse de Karl Marx de la technologie comme du travail solidifié (gelé), Geoffrey Bowker et Susan Leigh Star considèrent le logiciel comme un *discours d'organisation solidifié* (1997, p.284). Le point commun à ces approches est d'encastrier (*embedded*) des connaissances dans un support physique appropriable, reproductible et transportable indépendamment des agents humains. L'originalité par rapport à d'autres artefacts intégrant des connaissances comme l'écrit (livre, article) qui permettent simplement de décrire une procédure est qu'un logiciel permet de l'exécuter (Eric Brousseau, 1993, p. 207). De ce fait les compétences qui peuvent être transformées en marchandises ne se limitent plus à des savoirs mais englobent des savoir-faire, des concrétisations de l'application des connaissances. Une autre conséquence est qu'elle peut encore étendre la rationalisation de ces savoir-faire incorporés à des artefacts, par leurs possibilités de modifications et de manipulations supérieures à celles des êtres humains (Abbe Moshowitz, 1997, p. 31).

Nous verrons toutefois, par une analyse du logiciel en termes de caractéristiques, que dans l'exploitation des possibilités des logiciels, les compétences des utilisateurs gardent une place importante. Auparavant, il est nécessaire d'analyser brièvement ce qu'est la production d'un logiciel.

Section IV - La production d'un logiciel

Produire un logiciel consiste à traduire une représentation d'une partie de la réalité dans un langage compréhensible par une machine. "L'expression de cette pensée, à travers les différents langages de l'informatique, n'a de vérité que par rapport à la réalité dont elle constitue un modèle. Vis à vis de cette réalité, le langage informatique va donc fonctionner comme un métalangage permettant la génération d'entités logiques (termes primitifs, axiomes, définitions, termes composites, etc.) dont la composition, au moyen de procédés de construction quasi algébriques, constituera précisément une modélisation de la réalité. Le programmeur est donc soumis à tous les aléas de l'activité langagière : ambiguïté, paradoxes, inconsistance, incomplétude, indécidabilité, indéterminisme" (Jacques Printz, 1998, p. 234).

La production d'un logiciel peut donc être assimilée à une opération de codification, qui repose sur trois pôles : un modèle, une technologie de codification et un langage (Robin Cowan et Dominique Foray, 1998, p. 308). L'étude de l'économie de la codification fait apparaître qu'une décomposition de la codification en différentes étapes permet de réduire le coût de la codification, l'output d'une étape servant d'input à l'étape suivante (idem, p. 311). C'est ce processus qui s'est produit dans la production des logiciels avec la croissance du nombre d'étapes, qui consiste en des traductions successives d'un code de départ qui doit être le plus compréhensible possible par des humains (y compris non informaticiens) jusqu'à un code d'arrivée qui doit être compréhensible par la machine : par exemple les spécifications du problème à résoudre exprimées dans un langage graphique, la résolution de ce problème dans un langage de haut niveau (code source) et sa traduction automatique par un compilateur dans un langage machine (code objet).

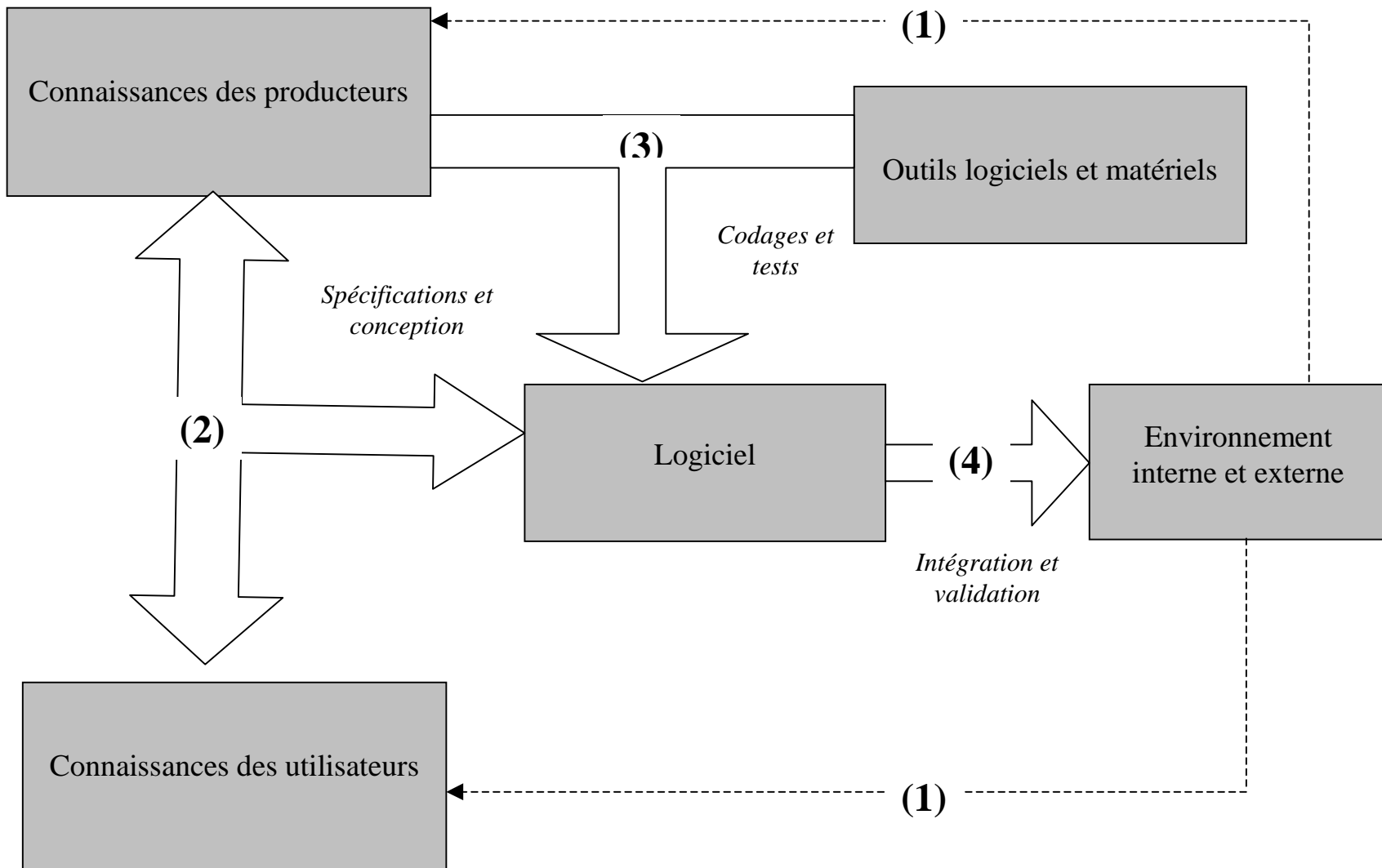
La décomposition du processus, outre qu'elle facilite le dialogue entre les utilisateurs et les concepteurs du logiciel, permet également d'isoler la résolution du problème auquel répond le logiciel. Elle permet notamment de séparer une phase d'analyse, durant laquelle on s'intéresse au *quoi*, à savoir les besoins du client ou de l'utilisateur, d'une phase de conception, durant laquelle on s'intéresse au *comment*, à savoir la solution logique.

L'activité de programmation englobe l'écriture du programme (agencer des données, des algorithmes³³ et des contrôles afin de communiquer avec l'ordinateur), la documentation du programme (afin de communiquer avec les autres programmeurs), la vérification, la validation et les tests de ce que l'on a programmé, et enfin les anticipations des évolutions possibles du programme (Jacques Printz, 1998, p. 237). Un programme est fondamentalement le résultat d'un raisonnement, avec deux possibilités : le programmeur dispose d'un schéma logique du programme, sous la forme d'un programme existant ou d'un modèle dans une bibliothèque d'algorithmes qu'il va traduire pour la situation précise ; le programmeur ne connaît que les ensembles d'entrées possibles et de résultats attendus et il doit découvrir un algorithme assurant la transformation des entrées en sorties (idem, p. 91).

Un logiciel est donc principalement un travail intellectuel de production de connaissances codifiées, fortement basé sur les connaissances tacites et difficilement appropriables des concepteurs, et sur l'utilisation d'outils logiciels et matériels. La production de logiciels "résulte de l'agencement original que réalise le producteur entre sa base de connaissances, en particulier dans le domaine technique considéré, les connaissances codifiées incorporées dans les logiciels auxquels il a accès et ce que sait ce producteur quant aux besoins qu'il doit satisfaire" (Marie de Besses, 1999, p. 264). Sans entrer dans le détail des différentes méthodes de production des logiciels, domaine en pleine évolution et qui sera analysé ultérieurement, on peut positionner les principales étapes de la production des logiciels (définition des besoins, spécifications du logiciel, conception générale puis détaillée, codage, tests, intégration, validation, et enfin maintenance) de la façon suivante (cf. schéma V).

³³ Un algorithme est une procédure de résolution précise de telle ou telle classe de problèmes nécessitant un ensemble déterminé d'opérations à effectuer sur un certain code, en un nombre fini d'étapes pour obtenir la solution du problème. Un algorithme est d'autant plus intéressant qu'il résout une large classe de problèmes. Un algorithme peut donner naissance à un programme informatique en le formulant dans le langage *ad hoc* de l'ordinateur et en tenant compte de toutes les limitations de la machine. En fait, beaucoup de programmes ne sont que des cas particuliers, sans intérêt du point de vue algorithmique, mais qui peuvent être très intéressants du point de vue pratique (Jacques Printz, 1998, p. 338).

Schéma V : La production d'un logiciel



Le schéma V met en évidence qu'un logiciel est destiné à s'intégrer à un environnement interne (le système informatique) et externe (son domaine d'application). Pour créer un logiciel, il faut réunir une connaissance du problème, une capacité à définir une solution et une capacité à construire la solution³⁴. La connaissance du problème repose sur les connaissances des producteurs et des utilisateurs, distinctes et complémentaires, du système informatique et du domaine d'application (1). L'intercompréhension entre les producteurs et les utilisateurs joue un rôle décisif dans l'élaboration des spécifications du logiciel et sa conception générale (2), d'autant plus qu'en général le mandat est complexe et confus, requérant une intervention active du mandant (Jacques Girin, 1994). Les formes concrètes de cette intercompréhension sont évidemment variables selon que le logiciel est fabriqué sur mesure pour quelques utilisateurs, qu'il est destiné à être utilisé par plusieurs milliers d'utilisateurs différents (un système de réservation par exemple) ou qu'il s'agit d'un progiciel vendu à des milliers d'exemplaires. Dans ces premières phases décisives durant lesquelles les erreurs sont les plus coûteuses à corriger, les outils logiciels et matériels ont un rôle secondaire. Par contre, les phases de codages et de tests (3) reposent de plus en plus sur la mise en œuvre par les producteurs d'outils de plus en plus sophistiqués (outils de génie logiciels) permettant une automatisation partielle. Il faut noter que la plupart de ces outils sont des logiciels, eux-mêmes conçus avec d'autres logiciels³⁵. Le logiciel une fois produit, il reste à l'intégrer dans son environnement (4) pour pouvoir valider la succession d'abstractions réalisées, par confrontation avec le concret qui avait servi de base à la modélisation (Bruno Latour, 1993).

Enfin, les compétences pour produire le logiciel reposent davantage sur la qualité des relations entre les utilisateurs, les producteurs et les outils mis à leur disposition que sur les propriétés de chaque élément pris séparément (Jacques Girin, 1994, p. 27). Les qualités du logiciel produit peuvent être appréhendées à partir d'une analyse en termes de caractéristiques.

³⁴ On retrouve la succession d'une tâche d'explicitation ou de mise en mots (réaliser une description) et d'une tâche de mise en actes (transformer une description en activité) caractéristique de toute activité (Jacques Girin, 1994, p. 17).

³⁵ Par exemple un outil de test est un logiciel qui a été produit à partir d'autres logiciels que sont des compilateurs, des éditeurs de textes, un système d'exploitation...

Section V - Une représentation du logiciel en termes de caractéristiques

Le logiciel se singularise par un nombre particulièrement élevé de caractéristiques pertinentes. Par exemple, pour apprécier la qualité des logiciels, la norme NF ISO/CEI 9126 publiée en 1991 (1992 pour la version NF) définit six "caractéristiques qualité" et 21 sous-caractéristiques (Philippe Robert, 1997, p. 168). Les six caractéristiques principales sont la *capacité fonctionnelle* (existence d'un ensemble de fonctions qui satisfont les besoins exprimés et implicites pour des tâches données), la *maintenabilité* (effort nécessaire pour diagnostiquer les déficiences ou les causes de défaillances, ou pour identifier les parties à modifier), la *facilité d'utilisation* (effort nécessaire pour l'utilisation et évaluation de cette utilisation pour un ensemble défini ou implicite d'utilisateurs), la *fiabilité* (aptitude du logiciel à maintenir son niveau de service dans des conditions précises et pendant une période déterminée), le *rendement* (rapport existant entre le niveau de service d'un logiciel et la quantité de ressources utilisées, dans des conditions déterminées) et la *portabilité* (aptitude du logiciel à être transféré d'un environnement à l'autre). Un des apports de cette norme est qu'elle met sur le même plan les caractéristiques fonctionnelles, c'est à dire ce que le système "fait ou est censé faire", et les caractéristiques non fonctionnelles, c'est à dire "comment c'est fait". Or, ces dernières caractéristiques, qui sont fondamentales pour apprécier le comportement concret d'un système informatique, étaient traditionnellement sous-estimées tant en termes d'importance que de moyens consacrés à leur amélioration, par rapport aux caractéristiques fonctionnelles considérées comme "sacrées" (idem, p. 169).

Parmi l'ensemble des caractéristiques, une distinction intéressante est celle qui a été effectuée entre les caractéristiques correspondant à des *critères de qualité* et celles correspondant à des *facteurs de qualité* (Serge Bouchy, 1994 ; Jean-Marc Geib, 1989). Les critères de qualité sont l'expression d'une *perception interne* du produit, représentant le point de vue du producteur et une mesure de la qualité de la fabrication ; les facteurs de qualité sont l'expression d'une perception externe du produit, représentant le point de vue de l'utilisateur et une mesure de sa satisfaction. Evidemment, le respect des critères de qualité doit être un moyen d'améliorer les facteurs de qualité.

On peut approfondir cette distinction en intégrant les apports de différents travaux sur les caractéristiques des produits. Il s'agit tout d'abord de l'étude de l'innovation dans les

services effectuée par Faïz Gallouj et Olivier Weinstein (1997). Cette analyse a pour point de départ les travaux de P.P. Saviotti et J.S. Metcalfe (1984). Pour analyser le changement technique, ces auteurs, à partir de la théorie de la consommation de K.J. Lancaster (1966), définissent et représentent un produit comme la mise en correspondance de trois ensembles de caractéristiques : des caractéristiques de process, des caractéristiques techniques et des caractéristiques de service (ou caractéristiques d'usage). L'application de cette analyse à l'étude de l'innovation dans les services conduit Faïz Gallouj et Olivier Weinstein à regrouper dans les caractéristiques techniques, les caractéristiques techniques et les caractéristiques de process (il n'est pas toujours évident dans le cas des services de séparer le produit du process), et à introduire les compétences des producteurs et des utilisateurs, donnant ainsi une plus grande importance au rôle des acteurs, producteurs et consommateurs, dans une analyse au départ assez "techniciste".

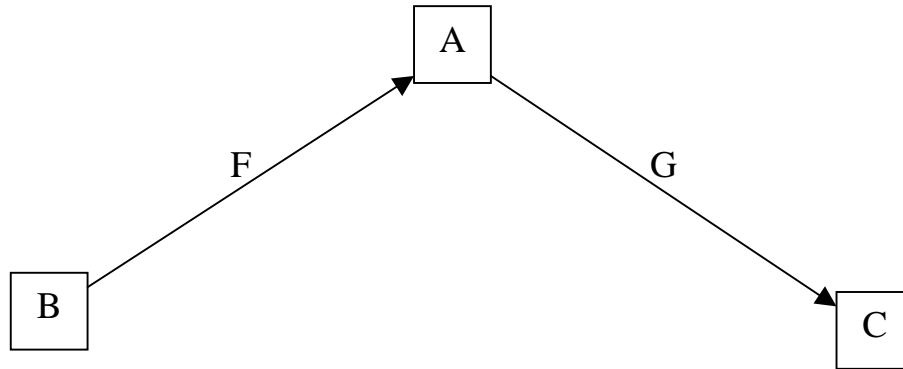
Le deuxième apport est constitué par les travaux de Jean Gadrey sur l'économie des services, qui insistent sur l'importance d'une distinction entre produit *immédiat* (ou direct) et produit *médiat* (ou indirect) d'une activité³⁶. Ceci nous conduit à distinguer dans les caractéristiques d'usage, des *caractéristiques d'usage directes* et des *caractéristiques d'usage indirectes*, ce qui permet d'insister sur le fait que les caractéristiques d'usage ne sont pas des données objectives immuables et parfaitement définies, mais des constructions sociales, des conventions sociales plus ou moins stables, permettant d'atténuer les différences de jugement et d'appréciation entre les acteurs concernés.

Le troisième apport est l'analyse du produit informatique de L.A. Gérard Varet et J.B. Zimmermann (1980), prolongée par les travaux de M. Delapierre et J.B. Zimmermann (1984, 1991). Le produit informatique y est considéré comme étant un *système*, constitué à partir d'une combinaison de composants et de sous-éléments. La définition du produit informatique "associe trois espaces, celui des biens informatiques élémentaires (...), celui des ensembles architecturés d'éléments en systèmes (...) et celui des modalités d'utilisation de ces systèmes"

³⁶ Ces produits sont qualifiés de *médiats* "parce qu'ils sont médiatisés par les individus bénéficiaires et parce qu'ils ne se manifestent pas immédiatement dans le temps" (Jean Gadrey, 1996 A, p. 95). Si "les rapports marchands ont généralement tendu à valoriser le produit immédiat, à en faire l'axe de la mesure, le centre de gravité de la marchandise et de son prix, (...) les changements en cours semblent indiquer une influence croissante des variables médiates, une prise en compte croissante des conditions d'usage dans la définition même du produit et de la marchandise" (idem, p. 135-136).

(M. Delapierre et J.B. Zimmermann, 1991, p. 10). Le produit informatique est représenté formellement ainsi :

Schéma VI
Le concept de produit informatique



Source : M. Delapierre et J.B. Zimmermann, 1994, p. 93

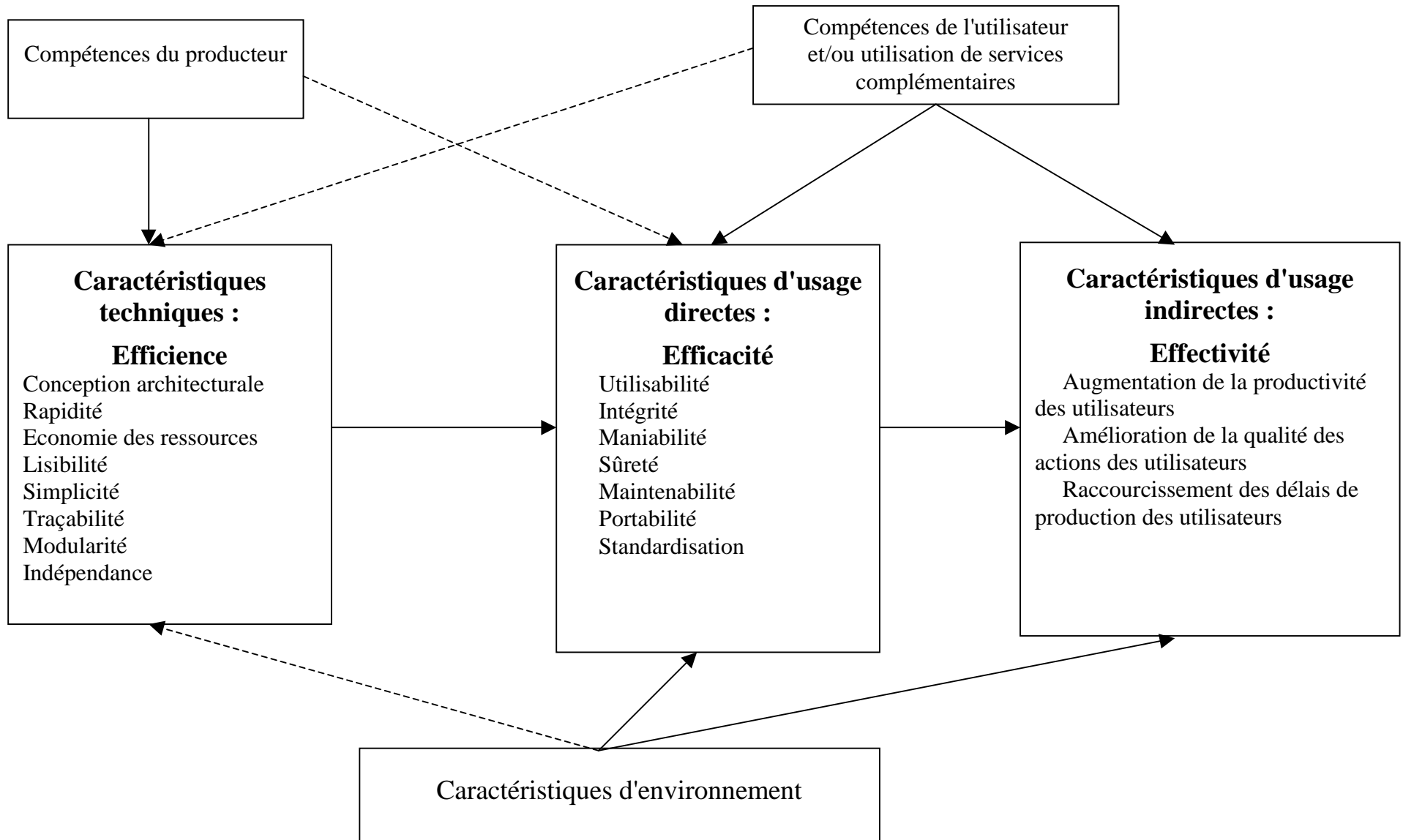
L'articulation entre les trois espaces constitutifs du produit informatique est représentée à travers la manière dont un complexe issu de l'un permet de réaliser un complexe de l'autre, par deux correspondances : une correspondance F de production des activités ou performances des systèmes (A), à partir des biens informatiques (B), qui résulte des capacités d'architecture des systèmes informatiques ; une correspondance G de production des caractéristiques d'utilisation (C) à partir des activités informatiques (A), qui traduit la capacité à mettre en œuvre les performances des systèmes en vue de la satisfaction de besoins d'utilisations spécifiques (M. Delapierre et J.B. Zimmermann, 1994, p. 92).

Nous intégrons ce troisième apport en ajoutant aux caractéristiques du logiciel, des caractéristiques d'environnement³⁷ dont l'importance vient du fait qu'un logiciel n'est qu'un composant d'un bien système³⁸, en l'occurrence un système informatique. La représentation proposée est la suivante :

³⁷ De façon plus générale, Herbert Simon souligne que pour un objet artificiel, "la réalisation d'une intention ou l'adaptation à un but implique une relation entre trois termes : l'intention ou le but, les caractéristiques de l'artefact et l'environnement dans lequel cet artefact est mis en œuvre" (1974, p. 20).

³⁸ Un "bien système" est la combinaison de plusieurs composantes élémentaires complémentaires. Les différentes composantes constituent des marchés séparés mais dont l'interdépendance est forte tant du côté de l'offre que du côté de la demande (Anne Perrot, 1995, p. 8).

Schéma VII : Les caractéristiques du logiciel



Certes, pour tous les produits, on peut distinguer des caractéristiques techniques et des caractéristiques d'usage directes et indirectes ; mais dans certains cas (normalisation des produits et de leurs caractéristiques techniques, stabilité ou prévisibilité de l'évolution de l'environnement, nécessité de compétences limitées de l'utilisateur), la connaissance des caractéristiques techniques suffit pratiquement à déterminer les caractéristiques d'usage. Les enchainements sont au contraire beaucoup plus complexes dans le cas des logiciels.

Le point de départ du schéma VII est constitué par les caractéristiques techniques du logiciel qui déterminent son *efficience*. C'est à partir de ces caractéristiques que d'autres informaticiens, examinant le code source du logiciel, formuleraient un jugement de qualité sur le logiciel considéré. Ces caractéristiques techniques résultent du processus de production du logiciel analysé précédemment et dépendent principalement des compétences des producteurs, des outils matériels et logiciels utilisés (non représentés sur ce schéma), mais également des compétences de l'utilisateur, notamment pour formuler ses besoins, et de la connaissance des caractéristiques de l'environnement dans lequel sera utilisé le logiciel.

Les caractéristiques d'usage directes du logiciel représentent le point de vue immédiat des utilisateurs, qui vont juger de son *efficacité* quand ils l'utilisent. Ces caractéristiques d'usage directes dépendent bien évidemment des caractéristiques techniques du logiciel, mais les compétences de l'utilisateur, vues sous l'angle de sa capacité à utiliser le logiciel, et les caractéristiques du système dans lequel est intégré le logiciel, jouent également un rôle important. Il faut notamment tenir compte des évolutions des compétences des utilisateurs (apprentissage) et du système informatique. Les compétences du producteur peuvent parfois conserver une influence directe (sans passer par la médiation de l'objet logiciel) quand le producteur du logiciel fournit également des services d'accompagnement (installation, formation, aide à l'utilisation, maintenance).

Les caractéristiques d'usage indirectes sont les véritables résultats de l'utilisation du logiciel et représentent *l'effectivité* du logiciel : il s'agit de savoir dans quelle mesure cet outil aide effectivement cet utilisateur en lui permettant d'améliorer sa productivité ou la qualité de son activité. L'efficacité de l'outil (appréhendue par ses caractéristiques d'usage directes) est une condition nécessaire mais non suffisante. Interviennent là encore les compétences de l'utilisateur, moins cette fois sous l'angle "informatique" que sous l'angle de la capacité à intégrer les services rendus par le logiciel à ses activités, et les caractéristiques de

l'environnement qui ne se limitent pas au système informatique mais intègrent l'ensemble du domaine d'application du logiciel considéré.

Clarifier les notions de données et de connaissances a permis de mettre en évidence quelques aspects importants de la codification des connaissances, qui seront utiles pour l'analyse de la production des logiciels, que l'on peut considérer comme une forme particulière de codification de connaissances. De même, l'analyse de l'information comme élément central d'un processus de communication de connaissances (direct ou médié), de par ses conséquences sur des questions importantes d'économie de l'information, constitue un apport intéressant pour l'analyse économique du logiciel.

L'étude des particularités du traitement automatique de l'information par des systèmes informatiques met en évidence la nécessité d'une intercompréhension forte entre concepteurs du système informatique et utilisateurs, en plus de l'intercompréhension entre émetteur et récepteur d'information, condition d'efficacité de tout processus informatif. Dans le traitement automatique de l'information, le logiciel a un statut original, objet frontière entre le monde humain et le monde machinique, "acteur non humain" partie prenante d'un système informatique. Un logiciel se présente sous la forme d'un texte numérique actif doté de nombreuses caractéristiques qui sont le reflet de la diversité des points de vue existants.

Un problème majeur est que certaines de ces caractéristiques peuvent être relativement contradictoires (par exemple, la rapidité et la fiabilité), ce qui nécessitera des arbitrages délicats sur les caractéristiques à privilégier. Une difficulté supplémentaire est que selon le type de logiciels (et pour un même type de logiciel selon le type d'utilisateurs) les caractéristiques pertinentes ne sont pas nécessairement les mêmes : par exemple, pour un simple usager seules importent les caractéristiques d'usage du logiciel, les caractéristiques techniques ne lui étant d'aucune utilité, hormis leur influence, qui lui est largement inconnue, sur les caractéristiques d'usage ; par contre, pour un utilisateur expert, qui joue un rôle plus important dans la construction des caractéristiques d'usage, la connaissance et la nature des caractéristiques techniques sont tout à fait décisives. Enfin, au sein de ces groupes de caractéristiques, la liste des caractéristiques décisives, sans parler du choix des indicateurs pour évaluer ces caractéristiques, est également variable.

Chapitre II

L'HISTOIRE DE L'INFORMATIQUE : LA CONSTITUTION

SUCCESSIVE DE RESEAUX TECHNICO-ECONOMIQUES

STRUCTURES PAR DES STANDARDS.

Une analyse de l'économie du logiciel ne peut se passer d'une analyse plus générale de l'économie de l'informatique pour plusieurs raisons liées. Tout d'abord parce qu'un logiciel est un composant d'un "bien système", en l'occurrence un système informatique (cf chapitre I) et qu'en conséquence, l'économie du logiciel fait partie de cet ensemble plus vaste qu'est l'économie de l'informatique dont elle "subit" les évolutions notamment technologiques¹. Ensuite parce que l'on ne peut comprendre le rôle d'un certain nombre d'acteurs majeurs de l'économie du logiciel en faisant abstraction de leur rôle dans le reste de l'économie de l'informatique. En particulier, il est intéressant de comprendre comment on est passé d'une situation où l'économie du logiciel était dominée par des acteurs informatiques dont l'activité principale était autre que la production des logiciels (en premier lieu IBM) à une situation où ce sont des entreprises spécialisées sur la production des logiciels qui jouent un rôle majeur sur l'évolution de l'ensemble de l'informatique. Enfin, parce qu'un certain nombre des mécanismes économiques indispensables pour analyser l'économie de l'informatique (externalités de réseaux, rendements croissants d'adoption, effets "feedback", verrouillage technologique...), jouent également un rôle décisif dans l'économie des logiciels.

Le secteur informatique (du matériel et du logiciel) malgré sa jeunesse a connu des changements structurels importants. L'informatique se caractérise par une incertitude extrême

¹ Nous avons mis "subit" entre guillemets pour souligner qu'un des enseignements de l'analyse de l'histoire de l'informatique est que, de plus en plus, c'est l'évolution des logiciels qui joue un rôle déterminant dans l'évolution de l'ensemble de l'informatique.

sur son évolution, incertitude de nature plus socio-économique² que technique. Sur le plan technique, la fréquence des innovations ne doit pas masquer l'inertie technologique qui caractérise certaines dimensions importantes de l'informatique : par exemple, tous les ordinateurs reposent depuis l'origine sur les principes architecturaux énoncés par Von Neuman et la grande majorité des applications informatiques utilisées actuellement ont été écrites en Cobol. Sur le plan économique, l'histoire de l'informatique se singularise par l'importance stratégique que peut très rapidement acquérir une entreprise dont la position ne semble plus pouvoir être remise en cause³, et par l'apparition périodique de nouveaux acteurs dont certains réussissent à jouer un rôle majeur dans l'évolution de l'informatique. La meilleure illustration de cet agencement particulier a été la domination sur le plan mondial de l'ensemble de l'informatique par IBM, ce qui ne s'était jamais produit à un pareil niveau dans aucun secteur de l'économie, mais qui n'a pas empêché la naissance et le développement extrêmement rapide de Digital Equipment avec la mini informatique, de Sun avec les stations de travail, d'Intel avec les microprocesseurs, de Microsoft avec les progiciels, de Compaq et d'Apple avec les micros, de Netscape, d'AOL ou de Yahoo avec le développement d'Internet. Pour expliquer avec le même cadre d'analyse ces phénomènes apparemment contradictoires, il est nécessaire de prendre en compte *simultanément* les innovations techniques, les comportements stratégiques des producteurs mais aussi les changements concernant la nature des principaux utilisateurs, les types de problèmes qu'ils souhaitent résoudre, et les modifications de leurs attentes, liées à des évolutions culturelles et sociales plus générales. Si chacun de ces aspects a fait l'objet de recherches technologiques, économiques, sociologiques séparées, il n'existe pas, à notre connaissance, de tentative d'intégration dans une approche unifiée de l'ensemble de ces dimensions, ce que nous nous proposons de faire à partir de la notion de réseau technico-économique. L'objectif n'est pas seulement de comprendre les évolutions passées mais également de cerner les enjeux importants des changements structurels actuels.

Après avoir défini la notion de réseau technico-économique, puis ses particularités dans le cas de l'informatique (section I), ce cadre conceptuel est mobilisé pour analyser l'histoire de

² Parmi les multiples exemples d'innovations technologiques qui semblaient promises (à tort) à un développement rapide, on peut citer, pour la période récente, le cas des technologies *push* destinée à "pousser" les informations vers des utilisateurs ciblés, et le cas des micro-ordinateurs de réseaux (NC).

³ Les performances boursières de ces entreprises sont le reflet de l'appréhension de ces situations.

l'informatique (section II) ainsi que la situation actuelle et les perspectives d'évolution probable (section III).

Section I - Réseaux technico-économiques et informatique

A - LES RESEAUX TECHNICO-ECONOMIQUES

Notre conception des réseaux technico-économiques est largement inspirée des travaux de Michel Callon.

Premièrement, elle tente d'intégrer les deux approches existantes dans les travaux associant réseau et dynamique technologique : d'une part, les approches de la diffusion et de la compétition technologiques fondées sur l'existence d'externalités de réseaux et de rendements croissants d'adoption et de phénomènes d'interconnexion, développées par B. Arthur (1988), P. David (1985), M. Katz et C. Shapiro (1985), J. Farrell et G. Saloner (1986) ; d'autre part, les approches de l'organisation des relations inter-firmes en réseau dont les fondements renouvellent la problématique des choix d'internalisation - externalisation des activités technologiques des firmes (Nadine Massard, 1997, p. 336). L'objectif est de combiner ces deux conceptions en intégrant dans l'analyse des réseaux d'entreprises une appréhension des relations d'acteurs plus riche que la simple opposition interne-externe : en effet, des alliances entre entreprises concurrentes ou complémentaires peuvent contribuer à la conception et au développement d'un réseau technologique ; symétriquement, l'existence d'un réseau technologique peut aider à la création de réseaux d'entreprises entre acteurs ralliés au même choix technologique. Dans les deux conceptions de la notion de réseau, celui-ci est "une architecture qui engendre une stabilité de l'environnement de l'ensemble des éléments qui le constituent" (Pierre Garrouste et Sylvie Gonzalez, 1995, p. 136).

Deuxièmement, un réseau est constitué "d'un ensemble d'entités humaines et non humaines individuelles ou collectives (définies par leur rôle, leur identité, leur programme...) et des relations dans lesquelles elles entrent" (Michel Callon, 1991, p. 204). La conception énoncée par Michel Callon, de façon quelque peu provocatrice, qui consiste à considérer que les acteurs d'un réseau peuvent être humains mais aussi non humains semble assez judicieuse dans l'informatique, où des objets matériels ou immatériels (ordinateur, voire même certains de ses composants - microprocesseur, système d'exploitation, protocole de communication...)

semblent s'être affranchis de leurs concepteurs en se dotant de capacités d'action autonomes, et contraignent les acteurs humains à "négocier" avec eux (cf. chapitre I).

Troisièmement, pour que ces acteurs hétérogènes puissent entrer en relation il est nécessaire que s'opèrent des *traductions*, "opérations de redéfinition de l'identité, des intérêts et des buts des entités humaines et non humaines qui se trouvent liées dans une même configuration organisationnelle" (Pierre-Benoît Joly et Vincent Mangematin, 1995, p. 44). La traduction "établit une équivalence toujours contestable entre des problèmes formulés par plusieurs acteurs dans des répertoires différents (...) ; en mettant en équivalence ces deux problèmes, l'opération de traduction identifie et définit les différents acteurs humains et non humains concernés par la formulation de ces problèmes et par leur résolution" (Michel Callon, 1989, p. 81). La réussite des opérations de traduction va permettre, par la convergence des points de vue et des intérêts des acteurs, l'émergence d'un réseau grâce à l'existence *d'intermédiaires* divers (textes, artefacts techniques, êtres humains et leurs compétences, monnaie...), qui désignent "tout ce qui passe d'un acteur à un autre et qui constitue la forme et la matière des relations qui s'instaurent entre eux" (Michel Callon, 1991, p. 197) ; un acteur est donc également un intermédiaire mais auquel "la mise en circulation d'autres intermédiaires est imputée" (Michel Callon, 1991, p. 206).

La constitution d'un réseau peut donc se résumer à la création "d'une chaîne de traduction [qui] se construit dans la controverse, [qui] est le produit d'une histoire qui en forme le contenu, [et qui] est (...) "dépendante de la trajectoire" (*path dependent*)" (Michel Callon et Bruno Latour, 1991, p. 33). Pour que cette épreuve fondatrice soit surmontée, il faut réussir à "*intéresser* d'autres acteurs qui vont s'allier à vous, à travers le dispositif que vous leur proposez, et pour les intéresser il faut accepter de traduire leurs demandes, attentes et observations dans le dispositif sous la forme de choix techniques appropriés" (Michel Callon, 1994 A, p. 12-13).

Pour stabiliser le réseau, il faut "*aligner*" le comportement des différents acteurs, ce qui est tout à la fois la cause et la conséquence de *l'irréversibilisation* du réseau. En effet, l'émergence d'une traduction, qui au départ n'est qu'une option ouverte parmi d'autres, crée de l'irréversibilité en rendant difficile le développement d'autres traductions concurrentes et en prédéterminant les traductions à venir (restriction de l'espace des possibilités). "L'irréversibilité croît à proportion que des effets de systèmes se créent dans lesquels chaque élément traduit, chaque intermédiaire, chaque traducteur s'inscrit dans un faisceau

d'interrelations : modifier un élément, c'est à dire le définir différemment, suppose que l'on s'engage dans un processus de retraduction généralisée (...). Plus les interrelations sont multiples et croisées, plus les éléments associés sont nombreux et hétérogènes (non humains, humains, conventions...), plus la coordination est forte et plus la probabilité de résistance des traductions est élevée" (Michel Callon, 1991, p. 219). Se déploie un processus d'autorenforcement, basé sur l'existence de rendements croissants d'adoption et d'externalités de réseaux, qui aboutit à une situation de *lock-in* sur une trajectoire technico-économique *path dependency*. La construction de cet environnement socio-technique peut prendre du temps mais, une fois établi, il aboutit à un réseau technico-économique qui évolue de manière relativement autonome, paraissant doté d'une volonté de reproduction et de développement (Sandra Braman, 1997, p. 106). Une telle situation d'irréversibilisation, caractérisée par un accroissement des liens internes au détriment des liens externes et par un renforcement des frontières entre le réseau et son "environnement", s'accompagne "généralement de la production de normes et de standards qui homogénéisent les comportements des acteurs et des techniques" (Michel Callon, 1992, p. 315).

Toutefois cette irréversibilisation n'est que relative ; au-delà des *reconfigurations restreintes* qui se caractérisent par un travail routinier de consolidation, d'amélioration continue et obstinée et qui voient les connexions se diversifier, l'irréversibilité augmenter et les rendements croître, existe la possibilité d'une *reconfiguration élargie*, plus improbable et plus radicale ; celle-ci, aboutissant à la constitution d'un nouveau réseau technico-économique, nécessite la création d'un nouvel espace de circulation qui doit être entièrement configuré : il faut convaincre, traduire des intérêts parfois contradictoires, créer des technologies compatibles, établir des infrastructures, former des spécialistes et parfois reconfigurer la société dans son entier (Michel Callon, 1993).

C'est cette dynamique de la création et de l'extension des réseaux technico-économiques, de leurs relations de coopération et de compétition (Michel Callon, 1994 A, p. 17) que nous voudrions étudier concrètement dans le cas de l'informatique.

B - L'APPLICATION DE LA NOTION DE RESEAU TECHNICO-ECONOMIQUE A L'INFORMATIQUE.

L'histoire de l'informatique peut ainsi s'analyser comme la constitution successive de réseaux technico-économiques cherchant à se développer. Pour étudier cette histoire, il est

nécessaire d'examiner comment la notion de réseau technico-économique se concrétise dans l'informatique.

A un premier niveau se trouve le réseau élémentaire constitué par un acteur humain et un ordinateur. A l'origine il n'existe pas d'intermédiaire entre eux et l'informaticien utilise le langage machine pour pouvoir traduire les opérations qu'il souhaite en "instructions" exécutées par l'ordinateur. Ce réseau va s'étendre avec la multiplication des intermédiaires entre l'homme et la machine ; langage d'assemblage, puis langage de programmation de haut niveau permettant d'écrire du code-source qui est converti en langage machine (code objet) par des assembleurs, interpréteurs ou compilateurs, création de systèmes d'exploitation et surtout d'applications très diverses répondant aux différents besoins des utilisateurs. Dès lors le réseau se complexifie en mettant en interrelation des acteurs humains divers (utilisateurs, concepteurs d'applications, concepteurs d'outils logiciels utilisés par les précédents...) et des acteurs non humains (unité centrale de l'ordinateur, périphériques divers, dispositifs d'interconnexion avec d'autres ordinateurs, objets non informatiques mais reliés à un ordinateur par des capteurs et/ou des actionneurs...). Entre ces différents acteurs mobilisés doivent s'opérer de multiples opérations de traduction dont la réussite conditionne l'existence et la viabilité de l'ensemble⁴.

Dans le même temps se constituent de multiples réseaux autour de chacun des intermédiaires. En effet, "chaque intermédiaire décrit et compose à lui tout seul un réseau dont il est en quelque sorte le support et l'ordonnateur" (Michel Callon, 1991, p. 199). Concrètement les acteurs qui utilisent un même microprocesseur ou un même système d'exploitation ou un même langage de programmation ou une même application, forment un réseau, et l'informatique peut être représentée par un enchevêtrement de multiples réseaux concurrents ou complémentaires. La réduction du nombre de réseaux concurrents s'opère par la standardisation de l'intermédiaire concerné, la standardisation pouvant se définir comme étant la "production des compatibilités nécessaires à l'exploitation des externalités de réseau afin de pouvoir bénéficier des avantages issus de l'intégration des réseaux" (Maryline Filippi, Emmanuel Pierre et André Torre, 1996, p. 91). La standardisation peut s'effectuer à de multiples niveaux : microprocesseur, matériel, système d'exploitation, outils logiciels, langage

⁴ C'est cet alignement que provoquent les multiples opérations de traduction pensées par d'autres que refusent les *hackeurs* en recherchant la communication directe avec le cœur de l'ordinateur en langage machine (Nicolas Dodier, 1995, p. 234)

de programmation, application, interface utilisateur (Pamela Gray, 1993, p. 37). Les différents intermédiaires n'ont toutefois pas la même importance sur l'évolution de l'ensemble du secteur informatique, les intermédiaires qui jouent un rôle stratégique étant différents selon les périodes de l'histoire de l'informatique. La nature de l'intermédiaire standardisé, et surtout les caractéristiques du processus de standardisation (modalités concrètes, rôle des différents acteurs...) vont avoir une influence décisive sur la structuration de l'économie de l'informatique⁵.

C - LA QUESTION DECISIVE DE LA STANDARDISATION

Un certain nombre de travaux économiques ont étudié les questions de standardisation en s'appuyant notamment sur des expériences historiques. Après avoir dégagé quelques enseignements majeurs de ces travaux (1), nous verrons les particularités des questions de standardisation dans l'informatique (2).

1 - Diversité des processus de standardisation et standardisation optimale

a - Les externalités de réseaux comme facteurs d'une standardisation nécessaire et difficile

L'établissement d'un standard est un cas particulier de compétitions entre solutions techniques. Brian Arthur (1988), à partir de la notion de rendements croissants d'adoption, a mis en évidence les mécanismes "d'autorenforcement" qui se créent autour d'une technologie : l'action même d'adopter une technologie rend celle-ci plus attractive pour les utilisateurs potentiels, augmentant par-là même ses chances d'être adoptée dans le futur. Les sources des rendements croissants d'adoption sont l'apprentissage par l'usage, les externalités de réseaux, les économies d'échelle en production, les rendements croissants d'informations et les interrelations technologiques.

Dans le cas des standards ce sont principalement les externalités de réseaux qui sont à la base des rendements croissants d'adoption (Dominique Foray, 1990). Les externalités de réseau⁶ sont des externalités de consommation qui proviennent de l'interdépendance des

⁵ "L'informatique finit inévitablement par tourner autour de la notion de standard" (Datamation, 1/8/93, p. 72).

⁶ Parfois également nommées effets de réseaux ou économies d'échelle liées à la demande.

décisions de consommation individuelle qui fait que la valeur d'un bien ou d'un service change quand il est acheté et consommé par d'autres utilisateurs : "on se trouve en présence d'un phénomène d'externalités de réseau lorsque les consommateurs d'un bien peuvent en tirer une jouissance d'autant plus grande que celui-ci est adopté par un nombre important d'usagers" (Jean Tirole, 1998, p. 390). Ces externalités de réseaux peuvent être directes (effet de club direct) comme dans les cas du téléphone, du fax ou du courrier électronique où l'existence d'un nouvel abonné augmente pour chaque usager l'utilité de son propre appareil ; si elles sont généralement positives, il faut mentionner la possibilité d'externalités négatives en cas de saturation d'un réseau physique de communication où l'arrivée de nouveaux automobilistes ou de nouveaux "internauts" peut faire baisser la valeur du bien ou du service pour l'ensemble des utilisateurs (Patrick Cohendet, 1996). Les externalités de réseaux indirectes reposent sur le fait que l'importance de l'offre de produits complémentaires dépend de la taille du réseau (exemple des magnétoscopes d'un standard donné et de la diversité des films proposés pour ce type d'appareil) ; le réseau est constitué par les possesseurs du produit principal qui ne sont pas nécessairement reliés entre eux au sens physique du terme (M. Katz et C. Shapiro, 1985)⁷. La force des externalités de réseaux positives aboutit à ce qui est connu sous le nom de *loi de Metcalfe*⁸ : cette loi, qui est plus une systématisation approximative d'observations empiriques qu'une loi au sens propre, stipule que la valeur d'un réseau, réel ou virtuel, augmente comme le carré du nombre de ses utilisateurs.

Les externalités de réseaux sont cependant différentes des autres rendements croissants d'adoption (Dominique Foray 1989). En effet, dans le cas général, la valeur de la technologie ne change pas pour l'utilisateur une fois qu'il l'a adoptée ; les critères d'adoption d'un utilisateur dépendent seulement des comportements d'adoption passés des autres utilisateurs (le processus d'adoption est uniquement "*path-dependent*"). Par contre dans le cas des externalités de réseaux, les rendements associés à une technologie sont déterminés par les comportements passés et *futurs* des autres utilisateurs, le processus d'adoption est "*path-and-future-dependent*" (Dominique Foray, 1990, p. 122). Dans ce cas, "l'anticipation constitue

⁷ De ce fait, les externalités de réseaux existent pour beaucoup de produits (l'automobile par exemple), mais la spécificité des biens d'information c'est que le produit peut ne plus avoir aucune valeur (Carl Shapiro, Hal R. Varian, 1999, p. 163).

⁸ Bob Metcalfe après avoir été l'inventeur au Xerox PARC, à la fin des années soixante dix, d'Ethernet, un standard de transmission de grosses quantités de données à grande vitesse développé pour les imprimantes laser, est le fondateur de la société 3Com, une entreprise qui propose des solutions de réseaux.

l'élément fondamental du choix de l'utilisateur potentiel" (Dominique Foray, 1990, p. 124), l'utilisateur doit adopter la technologie qui l'emportera à terme mais il peut être extrêmement difficile de prédire quelle sera cette technologie : "certaines structures dynamiques ne généreront jamais de séries temporelles assez longues, pour que les agents concernés puissent former des estimations probabilistes robustes sur les futurs possibles " (Dominique Foray et Christopher Freeman, 1992, p.16).

L'importance des anticipations explique que malgré la force des rendements croissants d'adoption, il ne se produit pas nécessairement une standardisation spontanée. On retrouve avec la production des standards, en raison de l'existence des externalités de réseaux, les problèmes ayant trait à la production des biens collectifs (Paul A. David, 1994).

b - Les standards peuvent être considérés comme des biens collectifs particuliers

Un standard définit précisément un procédé technique. Il constitue un langage commun qui permet d'assurer la production, l'échange mais aussi la compatibilité de la production des biens et services (Eric Brousseau, 1993, p. 203). Un standard correspond à une technique, un produit, une pratique qui est utilisée par une forte proportion d'agents, étant donné le nombre d'utilisateurs potentiels (Dominique Foray, 1996, p. 257). La notion de standard représente des réalités très diverses : des caractéristiques simples qui ont souvent un aspect relativement arbitraire (écartement des voix ferrées), des codes divers plus ou moins élaborés (depuis le Morse ou le code ASCII jusqu'aux protocoles de communication ou aux langages de programmation), des spécifications plus complexes de dispositifs techniques qui peuvent être immatériels (format de fichier, format de stockage de données) ou matériels (caractéristiques des interfaces), voire ces dispositifs techniques eux-mêmes (logiciels ou matériels). On constate que la standardisation peut concerner seulement des spécifications d'interface en désignant des fonctionnalités mais sans spécifier le produit (système ouvert reposant sur des standards "non-proprétaires") ou porter sur un produit spécifique qui devra être adopté (standard "propriétaire") (Jean-Benoît Zimmermann, 1995 B).

Ce continuum recouvre une évolution des formes de propriété depuis des biens libres, où la valeur sociale du bien vient uniquement de son adoption, jusqu'à des possibilités de breveter ou de protéger par un droit d'auteur le standard ou le support matériel ou logiciel sur lequel il repose. La frontière entre ces deux situations dépend de la nature du standard mais est aussi l'enjeu de controverses juridiques célèbres depuis les brevets déposés sur le morse

jusqu'aux possibilités de protéger l'utilisation d'une icône représentant une corbeille sur un écran d'ordinateur.

Les standards peuvent être considérés comme des biens collectifs. Dans le cas où les standards portent sur des caractéristiques inappropriables, il s'agit de biens collectifs purs dont on retrouve les déterminants d'indivisibilité (les dépenses de mise au point du standard sont indépendantes du nombre d'utilisateurs), de bien non-rival (le standard ne se détruit pas dans l'usage et peut donc être adopté par un nombre infini d'utilisateurs), et de non-exclusion de l'usage (on ne peut empêcher un utilisateur d'adopter le standard) (Dominique Foray, 1995). Dans le cas où il est possible de protéger le standard, il s'agit de biens collectifs mixtes avec externalités (Yves Crozet, 1997) caractérisés par une indivisibilité partielle ; en effet, dans ces situations, d'une part, les systèmes de protection légale (brevets, copyright) nécessitent de fournir de l'information notamment sur les caractéristiques du produit ou du procédé qui sont souvent des éléments déterminants d'un standard (Olivier Weinstein, 1989) ; d'autre part, si le producteur peut vendre l'utilisation du produit ou du procédé concerné, il lui est impossible de facturer *ex-ante* (avant la réalisation du processus de standardisation) son hypothétique futur caractère standard, qui demeure donc un bien collectif ; certes, *ex-post* (une fois le processus de standardisation réalisé) le producteur pourra internaliser la valeur supplémentaire qui résulte du caractère standard qu'a acquis son produit⁹, mais cela suppose que le processus de standardisation (ou production d'un standard) ait été effectué ou que les utilisateurs anticipent le succès d'un tel processus ; comme le notent Dominique Foray et Christopher Freeman (1992, p. 18) "il n'y a pas matière à différencier des phases de création et de diffusion : c'est l'adoption et l'usage qui confèrent au bien son mode d'existence. (...) Le processus de création recouvre dans ce cas la constitution du réseau, non pas la mise au point de l'artefact".

Concrètement lorsqu'apparaît un nouveau produit ou une nouvelle technique et que donc différentes solutions peuvent apparaître aussi légitimes ou performantes, la tentation est forte de chercher une "différenciation des produits et des méthodes de production, y compris par l'utilisation de stratégies créant des incompatibilités de réseau de façon à rendre certains

⁹ Cette possibilité n'existe que si le standard ne résulte pas d'un processus formel par un organisme de normalisation. En effet, lorsqu'une technologie est retenue dans le cadre d'un standard formel, l'entreprise qui l'a produit doit s'engager en contrepartie à licencier les brevets essentiels à l'adoption du standard "loyalement, raisonnablement et de façon non discriminatoire", le qualificatif de "raisonnable" correspondant à ce que le détenteur du brevet aurait obtenu dans un contexte de concurrence ouverte, par opposition à ce qu'il est en mesure d'extorquer une fois que le standard couvert par son brevet est adopté (Carl Shapiro, Hal R. Varian, 1999, p. 213).

consommateurs captifs" (Paul A. David, 1994, p.265). Dans ce type de situation, l'existence de barrières à la mobilité élevées augmente le pouvoir de marché de l'initiateur d'un système "propriétaire" (Eric Brousseau, 1993). C'est ce qui explique que "l'émergence de toute technologie de réseau (canaux, chemins de fer, télégraphe et téléphone) s'est accompagnée de la prolifération de systèmes concurrents incompatibles et, en conséquence, de l'impossibilité d'exploiter des externalités latentes de réseau" (OCDE, 1991 B, p. 40). Chaque producteur espère que c'est sa solution technique qui constituera le futur standard et on a de multiples exemples de coexistence durable de plusieurs techniques incompatibles : l'existence des grands systèmes informatiques qualifiés de "propriétaires", le demi-siècle nécessaire pour normaliser l'écartement des voies de chemin de fer en Angleterre (Dominique Foray, 1989), le réseau ferré australien sur lequel subsistent encore trois écarts de voies différents (Robin Cowan, 1995), la concurrence pour un standard entre 18 protocoles différents pour les programmes de télévision numérique par voie terrestre aux Etats-Unis (LMB Actu, 15/01/98)¹⁰.

Ces situations de sous-standardisation, quand elles perdurent, peuvent entraver le développement du secteur considéré ; on peut citer l'exemple de la dépression de l'industrie du disque après 1948 lorsque surgirent en même temps deux standards techniques nouveaux pour succéder au 78 tours : le 33 tours de CBS et le 45 tours de RCA, l'exemple des modems à haut débit (56 600 bauds) où une guerre des standards retarda leur adoption¹¹, ou l'exemple extrême de la guerre (suicidaire) de standards pour la radio stéréo AM qui empêcha l'émergence de cette technologie prometteuse (Carl Shapiro, Hal R. Varian, 1999, p. 232).

La crainte de devenir des "orphelins révoltés" (*Angry Technological Orphan*), c'est à dire de se retrouver piégé sur un mauvais choix technologique quand un standard s'imposera, peut aller jusqu'à l'absence de production comme l'illustre l'échec du son quadriphonique sur

¹⁰ On peut ajouter comme facteur secondaire des difficultés de standardisation, la présence d'acteurs économiques qui tirent leurs revenus de la coexistence de plusieurs standards : Carl Shapiro, Hal R. Varian cite le cas de Erie en Pennsylvanie où existaient trois écartements des rails différents, et où des rumeurs de standardisation provoquèrent des émeutes de la part des ouvriers spécialisés dans le chargement et le déchargement des wagons (1999, p. 186).

¹¹ Cette guerre qui opposait U.S. Robotics d'un côté et Rockwell/ Lucent de l'autre faillit déboucher sur une impasse au moment où la plupart des fournisseurs adoptait une technologie, alors que les utilisateurs privilégiaient l'autre technologie, d'où le risque que personne ne bénéficie de la vitesse de transmission accrue de ces modems. Finalement ces modems ne furent massivement adoptés qu'après qu'un standard international (V. 90) ait été édicté par l'ITU (Carl Shapiro, Hal R. Varian, 1999, p. p.235-237).

le marché grand public avec l'existence de deux standards concurrents en 1971, Columbia et JVC (Emmanuelle Le Nagard, 1997), ou les difficultés à s'imposer du DVD (Digital Video Disc) support réinscriptible destiné à remplacer les CD et CD-ROM. De même, l'importance des externalités indirectes peut bloquer le développement d'un produit par défaut de standardisation de celui-ci dans une situation de cercle vicieux ou "syndrome de l'œuf et de la poule" (Emmanuelle Le Nagard, 1997). Les débuts de la microinformatique étaient caractérisés par une telle situation : les différents types de micro-ordinateurs étaient incompatibles, générant des marchés segmentés et donc de taille trop limitée pour rentabiliser la production de progiciels diversifiés indispensables à leur utilisation par un large public, freinant ainsi le développement des micro-ordinateurs et donc la production des progiciels.

On retrouve ici typiquement les problèmes de production des biens collectifs : les agents économiques n'ont, chacun isolément, aucun intérêt à prendre une initiative qui serait pourtant profitable à chacun si tous la prenaient simultanément¹². Toutefois ceci n'implique pas que la seule solution pour atteindre une situation de standardisation soit le recours à la contrainte notamment étatique. En effet, pour que puisse s'enclencher un processus de standardisation, il suffit qu'il existe "une accumulation précoce de choix en faveur d'une même variante - même si ces choix ont été largement influencés par un ensemble de circonstances exceptionnelles et transitoires" (Paul A. David, 1994, p. 268-269). En effet, l'utilité pour un agent du ralliement à une variante constitue une fonction croissante du nombre d'agents ayant adopté cette variante. Dès que cette utilité dépasse les avantages escomptés d'une stratégie de différenciation, la variante considérée semble dotée d'un "magnétisme intrinsèque", jouer un rôle de "point focal" vers lequel convergent les comportements constituant une "convention auto-renforçante" (Robert Boyer et André Orlean, 1994, p. 220). Des comportements de "mimétisme rationnel" peuvent permettre l'existence d'équilibres sans autre raison que la croyance partagée dans leur existence : le nombre d'adhérents partageant le même point de vue impose celui-ci (Pierre-Yves Gomez, 1994). Les interactions dynamiques entre les utilisateurs ralliés à une solution et les utilisateurs potentiels, les *feedbacks* positifs qui en résultent (Paul A. David, 1994), font qu'une solution adoptée au départ par une proportion significative mais minoritaire des acteurs concernés peut devenir un standard. Le fait que le

¹² "La normalisation soulève les problèmes classiques de "bien public" : un coût de diffusion proche de zéro et des bénéfices qui ne peuvent être intégralement internalisés par les entreprises. Un sous-investissement privé est de ce fait à craindre et justifie une intervention publique" (Michel Catinat, 1998, p. 46).

nombre des adopteurs (réels ou potentiels) d'une solution technique dépasse un seuil critique permet de faire converger les anticipations vers cette solution, la transformant en standard *de facto*. Différents acteurs peuvent à la faveur de certains événements être à l'origine du déclenchement d'un tel processus.

c - Les différentes modalités de la standardisation

La première de ces modalités est l'action de l'Etat. Celle-ci s'exerce plutôt sur les caractéristiques des produits (standards-interface) que sur les produits eux-mêmes (standards-produits) (Jean-Benoît Zimmermann, 1995 B). Elle peut s'appuyer sur l'existence d'organismes de normalisation ou comités représentant les producteurs et parfois les utilisateurs concernés, que l'Etat peut soutenir et dont il peut reprendre les décisions en amplifiant l'effet d'annonce de leurs recommandations. L'Etat peut agir directement sur la standardisation en édictant des normes (standard *de jure*) qui s'imposent à l'ensemble des producteurs concernés. Dans certains cas cette intervention peut être très efficace. Par exemple, dans le cas des téléphones portables, la diffusion rapide de cette technologie en Europe s'explique par la décision administrative d'adopter un standard, le système GSM (*Global System for Mobile Communication*), alors qu'aux Etats-Unis l'approche libérale consistant à laisser faire le marché a abouti à la coexistence de trois technologies incompatibles et à un développement beaucoup plus faible du téléphone portable (Carl Shapiro, Hal R. Varian, 1999, p. 233). Mais l'Etat peut aussi agir indirectement et déclencher un processus de standardisation en tant qu'acheteur, en spécifiant des caractéristiques à respecter lors des appels d'offre pour les marchés publics (Robin Cowan, 1995). Cependant l'efficacité de l'intervention de l'Etat sur la standardisation peut être limitée : par exemple, dans le cas de la télévision couleur aux Etats-Unis, l'adoption par la FCC (Commission Fédérale des Communications) en 1950 du système CBS contre le système RCA à l'issue d'un test entre les deux systèmes, n'empêcha pas que sur le marché ce soit le système de RCA qui se soit imposé en raison de sa compatibilité avec la télévision noir et blanc (Carl Shapiro, Hal R. Varian, 1999, p. 191-192). Surtout l'action de l'Etat est plus problématique dans les cas de plus en plus fréquents où c'est le niveau international qui constitue l'espace pertinent de la standardisation. En effet, d'une part il est fréquent que les Etats utilisent des normes nationales spécifiques et des exigences de certification pour protéger les marchés intérieurs (Richard Hawkins, 1995), même si à terme il est difficile de conserver une norme nationale différente

d'une norme internationale qui se serait imposée (on peut citer l'exemple de l'abandon par la France à la fin des années 60 de sa propre technologie nucléaire au profit de la technologie qui dominait le marché mondial) (Robin Cowan, 1995). D'autre part, il est difficile pour les Etats d'agir sur une standardisation directement au niveau international : le temps nécessaire pour parvenir à s'accorder sur des spécifications communes en présence d'intérêts divergents - déjà important au niveau national - peut devenir rédhibitoire au niveau international, surtout en présence de techniques et de problèmes qui évoluent rapidement ; les institutions internationales comme le GATT, ou depuis 1995 l'OMC, ne disposent pratiquement d'aucun pouvoir pour faire respecter les décisions qu'elles adoptent (Richard Hawkins, 1995), et il n'existe pas d'équivalent des marchés publics au niveau international.

Le processus de standardisation peut également être enclenché par une entreprise (ou une alliance entre entreprises¹³) en position de force sur le marché du produit concerné voire sur un marché complémentaire. Dans le cas où la standardisation porte sur des caractéristiques non-appropriables (standard-interface) elle aboutit à la production de produits compatibles¹⁴ par les autres producteurs ralliés au standard. L'entreprise à l'initiative de la standardisation détient néanmoins un avantage sur les entreprises concurrentes dans sa capacité à faire évoluer le standard en fonction de ses intérêts. Dans le cas où il est possible pour l'entreprise de protéger par brevet ou copyright le produit ou le procédé qui fait l'objet du standard, l'entreprise peut en retirer un profit important soit parce qu'elle peut se retrouver en situation de monopole sur le marché, soit par la vente de licences aux autres producteurs. C'est ce qui explique que pour imposer un standard une entreprise peut recourir à différents moyens : baisse des prix (voire même distribution gratuite du produit) qui pourra être compensée par une hausse quand l'entreprise sera en situation de monopole technologique ou par des stratégies de rentabilité croisée sur des produits complémentaires, multiplication des effets

¹³ Carl Shapiro, Hal R. Varian signalent toutefois qu'aux Etats-Unis, une coopération avec des entreprises concurrentes pour l'adoption d'un standard, peut être considérée comme une violation des lois interdisant les cartels et les ententes (1999, p. 263).

¹⁴ Le terme compatible est ambigu ; il peut aussi bien désigner les possibilités d'intégration avec des produits complémentaires (une imprimante compatible avec un ordinateur MacIntosh) que le fait qu'un produit de par ses caractéristiques soit parfaitement substituable au produit qui fait l'objet du standard (un micro-ordinateur compatible IBM PC).

d'annonce¹⁵ et des alliances pour déclencher des anticipations autoréalisatrices qui se forment souvent de manière subjective (Emmanuelle Le Nagard, 1997).

Enfin indépendamment des cas précédents, il peut exister des situations, notamment lorsque le produit ou le procédé sont radicalement nouveaux, où de "petits événements" exogènes suffisent pour produire un effet de localisation du progrès technique sur une technologie particulière à partir des choix des premiers utilisateurs qui ont dans ce cas une extraordinaire importance (Dominique Foray, 1990).

Les différents cas précédents peuvent se combiner ; par exemple des circonstances particulières peuvent permettre à une entreprise de développer un standard qui peut être validé par un organisme de normalisation et légitimé par l'Etat. Mais dans tous les cas rien ne garantit que le standard qui s'imposera soit optimal.

d - La standardisation peut ne pas être optimale...

On peut analyser le choix d'un standard comme un processus de sélection d'un équilibre de coordination au sein d'une multiplicité d'équilibres possibles et rien ne garantit que l'équilibre auquel on parvienne soit le plus efficace (au sens de Pareto). Comme le montre Laurent Thévenot (1989) dans un cadre plus général, des conventions conduisent à choisir un équilibre parmi les possibles sans qu'on puisse juger de son optimalité. En effet, ce qui conduit le processus à converger vers un équilibre donné ce sont essentiellement des circonstances initiales historiques particulières. Par la suite les choix effectués par les agents qui adoptent la même solution ne sont pas seulement (ni même principalement) dictés par leurs préférences personnelles indépendamment des décisions des autres agents¹⁶, mais par les externalités résultant du choix effectué par d'autres agents en faveur de cette solution. En retour, le ralliement à cette solution augmente les externalités positives pour les autres agents. On comprend que ce type de processus "autorenforçant" puisse conduire à des situations de

¹⁵ L'importance des effets d'annonce peut conduire à des pratiques douteuses : dans l'affrontement entre U.S. Robotics et Rockwell/Lucent pour imposer leur technologie de modems à haut débit (cf. supra), chacun des deux camps affichait sur un site Web la liste des entreprises ayant choisi leur technologie. Une vérification effectuée par la magazine PC World montra que seule une minorité des entreprises citées offraient effectivement une connexion de ce type (Carl Shapiro, Hal R. Varian, 1999, p. p.236-237).

¹⁶ On est en présence d'un comportement de "*mimétisme stratégique*" où un individu copie les autres individus en considérant que "le fait qu'ils aient raison ou tort est secondaire" mais "qu'il est plus avantageux, en termes de risque, de rendement ou de réputation, d'être en accord avec la majorité" (André Orléan et Yasmina Tadjeddine, 1998, p. 187).

standardisation exagérée ou à une standardisation trop précoce (une fois le processus engagé, d'autres possibilités même plus performantes ne parviennent pas à s'imposer) ou à l'adoption de standards non-optimaux (on peut citer les exemples du VHS par rapport au Betamax dans la vidéo, et du clavier QWERTY contre les claviers Ideal ou Dvorak). Il se produit un "verrouillage en raison d'accidents historiques" (*accidental lock-in*, Paul A. David, 1994). "Ce n'est pas parce qu'elle est moins efficace qu'une technologie n'est pas choisie et dès lors disparaît, mais c'est parce qu'elle n'est pas choisie qu'elle devient moins efficace que sa rivale" (Dominique Foray, 1990, p. 119). Les solutions abandonnées pouvaient offrir des perspectives très supérieures, malgré des défauts de jeunesse résultant éventuellement de leur nature plus novatrice (OCDE, 1991 B). De même l'événement historique (technologique ou économique) qui avait enclenché le processus de standardisation peut ne plus être pertinent rendant le standard inefficace (la disposition du clavier QWERTY avait été choisie de façon à éviter la frappe de touches voisines qui emmêlaient les marteaux des machines à écrire)¹⁷.

e - ...et malgré tout perdurer

En effet, quand un standard existe, pour qu'une nouvelle solution, même jugée préférable par l'ensemble des utilisateurs, réussisse à s'imposer spontanément, il est indispensable d'une part, que les coûts résultant du changement de standard soient inférieurs aux bénéfices attendus de l'utilisation du nouveau standard, d'autre part qu'il existe une *information complète* au sein de la communauté des utilisateurs sur les intentions de chacun d'entre eux. Cette information complète est nécessaire pour que puisse s'enclencher un processus de "raisonnement à rebours" (*backward induction*) : le dernier utilisateur n'adoptera la nouvelle solution qu'une fois que tous les autres utilisateurs l'auront adoptée, l'avant-dernier utilisateur doit anticiper correctement le comportement du dernier utilisateur et ce processus d'anticipations correctes en chaîne doit exister jusqu'au premier utilisateur qui ne modifiera effectivement son comportement que s'il anticipe que tous les autres utilisateurs suivront son exemple. Il suffit donc que l'hypothèse d'une communication *parfaite* entre les utilisateurs ne soit pas totalement vérifiée (ce qui est le cas dès que le nombre d'utilisateurs dépasse un

¹⁷ De même l'écartement des rails de 4 pieds et 8 pouces et demi qui existait dans le nord des Etats-Unis, et qui finira par s'imposer dans tous les Etats-Unis avait été choisi conformément à la tradition anglaise, laquelle remontait à l'époque des Romains : elle correspondait à l'écartement optimal... d'un chariot en pleine charge tiré par un cheval (Carl Shapiro, Hal R. Varian, 1999, p. 186).

certain seuil) pour que ce processus ne puisse se dérouler¹⁸. Comme le note Paul A. David (1994, p.269), "le problème que posent ces communautés est que l'intercommunication y est assez développée pour engendrer des effets d'entraînements suffisamment puissants pour produire un consensus, même autour de conventions sous-optimales, mais qu'elles ont rarement atteint le degré extrême de communication sociale - la production d'une information complète sur les intentions de chacun des membres du groupe - qui permettrait de défaire ces conventions une fois qu'elles ont été établies".

On peut caractériser la convention qui se crée autour d'un standard comme relevant d'une stratégie évolutionnairement stable (Robert Boyer et André Orléan, 1994, p. 222-224). Ces auteurs reprennent le concept de stratégie évolutionnairement stable ou SES, proposé initialement par J. Maynard-Smith et G.R. Price, 1973) : "une stratégie telle que si elle est adoptée par tous les membres d'une population, aucune stratégie mutante ne pourra venir envahir cette population par les mécanismes de la sélection naturelle". Robert Boyer et André Orléan soulignent que dans le cas d'une SES, aucune autre stratégie ne peut s'imposer, même si cette nouvelle stratégie est elle-même une SES et qu'elle est supérieure à l'ancienne. En effet, un petit groupe qui suit une autre stratégie obtient de moins bons résultats et est condamné à disparaître progressivement. "Une telle situation d'excès d'inertie, qui peut donc se maintenir en dépit de préférences individuelles pour le changement, affectera principalement les marchés au sein desquels les externalités de réseau sont importantes et où donc un comportement de changement isolé ferait supporter à son auteur des coûts particulièrement élevés" (Dominique Foray, 1990, p. 126)¹⁹. Toutefois si les rendements croissants d'adoption proviennent principalement des externalités de réseau la situation de *lock-in* n'est pas irréversible (Dominique Foray, 1989) : une nouvelle convention peut émerger si elle est adoptée *simultanément* par une proportion d'individus dépassant un certain seuil critique²⁰.

¹⁸ Il est très difficile pour les utilisateurs de coordonner leur passage à une nouvelle technologie incompatible avec la précédente. Les coûts collectifs de changement peuvent être extrêmement élevés et ne sont pas linéaires : il est proportionnellement plus difficile de convaincre dix personnes appartenant à un réseau d'opter pour un autre réseau incompatible, que de convaincre une seule personne (Carl Shapiro, Hal R. Varian, 1999, p. 165 et 166).

¹⁹ Aucun utilisateur ne veut prendre individuellement le risque d'être le premier à perdre le bénéfice des externalités de réseau auquel il appartient et d'être ainsi marginalisé (Carl Shapiro, Hal R. Varian, 1999, p. 166).

²⁰ Pour une démonstration formalisée, cf. Robert Boyer et André Orléan (1994, p. 223-226).

f - La nécessaire évolution des standards : l'efficacité dynamique

Le raisonnement, basé sur les travaux de Dominique Foray (1990), est le suivant : "le processus d'élaboration d'un standard, en tant qu'activité de création, contient en lui-même des forces considérables d'inertie technologique". Or la standardisation dans le domaine des technologies de réseau intervient toujours à la fois trop tôt en bloquant le développement technologique et "trop tard car dès l'émergence de plusieurs réseaux incompatibles, des économies externes latentes ne sont pas exploitées". La logique de l'efficacité dynamique porteuse d'une évolution des standards est contradictoire avec la logique de l'efficacité statique qui vise à exploiter les externalités de réseaux en fixant les utilisateurs sur un standard déterminé. Quel que soit le mode de coordination (Etat ou marché) c'est toujours la recherche de l'efficacité statique qui est privilégiée. En effet, l'adoption d'une technologie procure des avantages à ses utilisateurs mais également des informations sur ces propriétés techniques. La standardisation par le marché ne prendra en compte que le premier aspect (Robin Cowan, 1995). Dans ce cas, "le marché se transforme en une puissante machine à fabriquer de l'irréversibilité et à restreindre la variété des options technologiques, c'est à dire l'espace des choix possibles" (Michel Callon, 1994 A). En théorie l'Etat pourrait mieux prendre en compte l'aspect générateur d'informations par l'allongement de la période initiale de compétition (avant le "*lock-in*") en favorisant systématiquement la variante la moins bien placée (Dominique Foray, 1990). Toutefois, "l'action de l'Etat ne pourra être menée efficacement que si elle intervient au tout début du processus, c'est à dire avant que les mécanismes d'autorenforcement n'enclenchent une dynamique de sélection échappant à toute régulation" (idem, p. 125). De même, la crainte d'avoir ultérieurement un nombre élevé d'"*Angry Techological Orphan*" incite les pouvoirs publics à privilégier la technologie dominante. En fait, "l'autorité centrale aura une capacité d'influence maximale au cours de la période durant laquelle elle ne possède qu'un minimum d'informations sur les avantages respectifs des technologies en compétition" (idem).

La question de l'évolution des standards (efficacité dynamique) est pourtant décisive dans le cas des technologies de l'information en raison de la rapidité des évolutions techniques et de l'élargissement de l'espace pertinent de la standardisation que ce soit sur un plan géographique ou sur le plan du rapprochement de technologies séparées (informatique, télécommunications).

2 - Les particularités de la standardisation dans l'informatique

L'informatique apparaît ainsi dans une situation particulièrement délicate par rapport à la standardisation. Les rendements croissants d'adoption sont très élevés dans l'informatique, que ce soient l'apprentissage par l'usage (dont l'importance dans le cas de l'informatique n'est plus à démontrer), les interrelations technologiques (entre les différents composants matériels et logiciels), les économies d'échelle en production (liées à la part déterminante des coûts fixes dans les coûts de production), les rendements croissants d'informations (importance de la connaissance des produits existants) et les externalités de réseaux. Celles-ci peuvent être directes en raison des connexions entre les ordinateurs (liaison physique ou dispositifs circulant entre les ordinateurs comme les disquettes) ou indirectes (la diversité de l'offre de produits complémentaires, comme les progiciels, dépend de la taille du réseau constitué par les possesseurs d'un ordinateur de type particulier).

De ce fait, les situations de "verrouillage", même si elles ne sont jamais absolues, sont particulièrement importantes en raison de l'ampleur des coûts de changements dues à la présence de nombreux "actifs durables, à la fois spécifiques et complémentaires : le remplacement d'un ancien système par un nouveau système implique souvent le changement de *tous* ses composants : données, équipements et capital humain" (Carl Shapiro, Hal R. Varian, 1999, p. 17). De plus les différents composants peuvent avoir des durées de vie très différentes (les logiciels ne "s'usent" pas !). Ces auteurs soulignent comment "les coûts de changement d'un système peuvent être astronomiques, alors même que le système le plus performant disponible actuellement sur le marché sera dépassé dans quelques mois" (idem). En particulier les coûts de transferts des données peuvent être prohibitifs sans compter qu'il n'est pas exclu que certaines informations soient perdues lors de cette opération. Enfin l'importance des apprentissages spécifiques explique que contrairement au cas de nombreux biens durables, dont le coût de remplacement diminue avec la dépréciation et le progrès technique, les coûts de changement augmentent avec le temps, à mesure que le personnel se familiarise avec un système dont l'utilisation efficace prend beaucoup de temps et avec lequel on devient de plus en plus performant avec l'expérience²¹.

²¹ Ce savoir-faire acquis dans l'utilisation du matériel et des logiciels constitue "le *wetware*" (Carl Shapiro, Hal R. Varian, 1999, p. 17).

La standardisation est donc à la fois de plus en plus indispensable vu l'importance croissante des phénomènes d'interconnexion dans l'informatique²², peut se réaliser très rapidement en raison de la vigueur des rendements croissants d'adoption et notamment des externalités de réseaux directes et indirectes, mais doit pouvoir garder un caractère évolutif pour répondre aux changements rapides et difficilement prévisibles des technologies informatiques.

Dans ce contexte la production de normes étatiques a une efficacité souvent limitée²³. Certes, les comités de normalisation ont tendance à se multiplier à tous les niveaux (OCDE, 1991 B), mais il est de plus en plus difficile de réaliser un consensus nécessaire à la prise de décision, vu la multiplicité des intérêts souvent contradictoires en présence. Dans le cadre de procédures longues²⁴, de plus en plus complexes et souvent opaques (OCDE, 1991 B), les instances de normalisation soit se contentent de jouer un rôle de "chambre d'enregistrement des décisions du marché" (Dominique Foray, 1990, p 133), les normes étant appliquées avant même que les décisions formelles soient prises, soit élaborent des normes fonctionnelles très générales à la portée pratique limitée : on peut citer l'exemple des normes fonctionnelles OSI (*Open-Systems Interconnexion*) ; ces normes concernant les réseaux informatiques définissent les performances à réaliser à sept niveaux (ou couches) de systèmes technologiques tout en laissant une large plage de liberté pour décider de la manière dont les normes seront respectées ; le choix d'un modèle de référence uniquement fonctionnel était lié au fait qu'IBM disposait depuis 1974 d'un ensemble de standards opérationnels définissant les spécifications de ses propres produits (SNA) permettant la normalisation interne de ses réseaux (Dominique Foray, 1990) ; dans la pratique, si la production des normes OSI augmente régulièrement depuis 30 ans leur impact reste limité : seules quelques-unes sont effectivement acceptées sur

²² De ce fait, la standardisation concerne également les formats des données qui circulent. Eric Brousseau cite l'exemple de l'échange des données informatisées (EDI) où existe un "archipel de normes sectorielles au sein de chaque espace national". Pour les entreprises qui réalisent des échanges avec des entreprises d'autres secteurs (c'est à dire quasiment toutes) et a fortiori pour les entreprises internationalisées, un standard universel serait plus efficient mais il a des difficultés à émerger en raison des stratégies de préservation de la spécificité de certains actifs et de la protection d'investissements existants (1993, p. 332).

²³ "L'industrie informatique a finalement retenu une leçon importante : la fixation des standards par des structures lourdes et lentes n'est pas adaptée à l'évolution rapide du marché" (Carl Shapiro, Hal R. Varian, 1999, p. 225).

²⁴ Eric Brousseau (1993, p. 271) cite l'exemple dans le domaine de l'Echange de Documents Informatisés (EDI), de l'Edifact Biard, instance de l'ONU chargée de définir les normes documentaires, qui a mis 15 ans (de 1975 à 1990) pour parvenir à la définition d'une première norme réellement opérationnelle (la facture).

le marché, de nombreuses normes sont inapplicables car se référant à des technologies périmées ou parce qu'elles se situent à un niveau trop théorique (Richard Hawkins, 1995), et rien ne garantit que deux machines respectant dans leurs grandes lignes les exigences d'une norme OSI acceptée répondront effectivement à la demande de compatibilité et d'interfonctionnement et pourront communiquer entre elles (OCDE, 1991 B).

En conséquence, les standards dans l'informatique sont plus souvent des standards *de facto* que des standards *de jure* (ou normes), et peuvent prendre la forme de standards "propriétaires" (standards privés)²⁵ pour désigner le contrôle d'un standard et de son évolution par une entreprise - ce qui lui confère un pouvoir majeur sur le marché - ou de standards ouverts (standards collectifs) quand le standard est l'œuvre d'un organisme alternatif (Pierre-José Billote, 1997, p.3). Des institutions *ad hoc*, issues principalement des milieux scientifiques et universitaires peuvent se révéler particulièrement efficaces pour promouvoir une standardisation dynamique. En effet, elles répondent aux conditions le plus souvent énoncées pour avoir des processus de standardisation satisfaisants : les standards doivent être des instruments de création de technologies et être élaborés par anticipation (OCDE, 1991 B), faire partie de la recherche-développement préconcurrentielle (Richard Hawkins, 1995) effectuée par des structures collectives de recherche coopérative (Dominique Foray, 1990). Ce qui peut permettre à ces institutions de développer des solutions ouvertes et évolutives, c'est leur plus grande souplesse de fonctionnement (par rapport aux institutions étatiques nationales ou internationales), et leur absence d'intérêt économique (par rapport aux entreprises privées) à figer les standards sur un plan temporel et/ou dans des produits définis. La diversité des acteurs qui composent ces institutions permet une grande variété des connaissances produites (Michel Callon, 1994 B) et donc une ouverture sur des solutions innovantes. Les traditions de communication et de publicité des travaux effectués garantissent une évolution rapide des standards qui s'efforce de maintenir la compatibilité avec les standards précédents en fonction des possibilités techniques, sans qu'interviennent des considérations de rentabilité économique privée. Ceci est remarquablement illustré par le rôle qu'ont joué ces institutions dans le développement d'Internet.

²⁵ Le terme "propriétaire" largement utilisé en informatique n'est pas toujours adéquat ("propriétaire de quoi ?" s'interroge Gérard Dréan, 1996 A, p. 40). Concernant des standards contrôlés par une entreprise, il semble plus judicieux de parler de standards privés (Pierre-José Billote, 1997, p.3). L'expression de standards "fermés" est parfois également utilisée par opposition à standards "ouverts" qui sont "des standards accessibles par tous et indépendants de leurs fournisseurs" (consortium Unix X/Open).

A l'obtention de la compatibilité reposant sur la standardisation (compatibilité *ex-ante*) est parfois opposée (P. David et J. Bunn, 1988) une compatibilité *ex-post* par des dispositifs de conversion (*gateways*) ; la première solution aurait pour inconvénient de générer une perte de variété tandis que la seconde induirait des coûts de conversion. Dans l'informatique, le recours à des adaptateurs (matériels ou logiciels) ne constitue pas véritablement une alternative à la standardisation. En effet, soit ces dispositifs sont relativement inefficaces et imparfaits, et la perte de performance qui en résulte n'empêche pas, malgré l'augmentation de la puissance du matériel, la nécessité à terme d'une standardisation²⁶, soit ces dispositifs sont relativement performants et dans ce cas ce sont ces technologies d'interface qui deviennent l'objet du processus de standardisation. On a ainsi assisté dans l'informatique à des déplacements du problème de la standardisation d'un niveau à un autre niveau : du matériel vers les systèmes d'exploitation (MSDOS puis Windows pour la microinformatique, UNIX pour mini-ordinateurs et les stations de travail), puis vers les communications entre des machines ayant des systèmes d'exploitation différents (TCP/IP pour les protocoles de communication, HTML pour la présentation des informations) et peut-être demain vers les applications qui pourront s'exécuter indépendamment du type de machines ou de systèmes d'exploitation (le langage JAVA).

Les standards sont donc les intermédiaires déterminants des réseaux technico-économiques en informatique, qui "alignent" des éléments très divers : des objets matériels (comme les microprocesseurs) et immatériels (comme les systèmes d'exploitation), des producteurs de matériels et de logiciels, des utilisateurs, des types de problèmes à résoudre, des valeurs culturelles. Ce réseau, une fois établi, semble se transformer en "être collectif", animé par une volonté de puissance, ayant pour objectif de survivre et de s'étendre.

Toutefois, dans un contexte marqué par la très forte croissance de l'informatique, de nouveaux réseaux technico-économiques peuvent surgir, ces nouveaux réseaux ne se substituant pas aux anciens mais coexistant avec eux. Pour que puisse émerger un nouveau

²⁶ Un exemple classique est l'émulation par un ordinateur du fonctionnement d'un ordinateur d'un autre type. Carl Shapiro et Hal R. Varian citent l'exemple des processeurs Alpha pour lesquels Digital a conçu un dispositif permettant de faire tourner les logiciels conçus pour Intel (dont le nombre est beaucoup plus important) mais qui, ce faisant, perd pratiquement toute la puissance supplémentaire qu'il avait par rapport aux processeurs d'Intel (1999, p. 251).

réseau, il faut qu'il y ait conjonction entre des innovations techniques²⁷, de nouveaux besoins, peu ou mal couverts par l'ancien réseau, et qui correspondent à des changements culturels (nouvelles valeurs, nouvelles attentes), de nouveaux utilisateurs et de nouveaux producteurs, qui n'ont pas à gérer les contraintes que représentent l'évolution de la base installée (Jean-Marie Desaintquentin et Bernard Sauter, 1991, p. 12-13). Il est également nécessaire qu'entre ces acteurs très hétérogènes une chaîne de traduction se crée permettant, par la convergence des intérêts et des objectifs, d'enclencher un processus de standardisation (qui le plus souvent déplace la question de la standardisation d'un intermédiaire à un autre), et que ce nouveau réseau n'apparaisse pas au départ comme s'affrontant directement aux intérêts des acteurs dominants de l'ancien réseau, pour éviter l'intégration du nouveau réseau par l'ancien (reconfiguration restreinte) en dénaturant son caractère innovateur. Ceci est rendu possible par la dynamique rétrograde (*backward looking*) dont font souvent preuve les entreprises informatiques dominantes : "en réaction aux changements de paradigme, l'action stratégique est mise en œuvre sur la base de représentations technico-économiques tournées vers le passé" (Godefroy Dang Nguyen et Denis Phan, 1997, p. 197)²⁸. Par contre, une fois que le nouveau réseau a commencé à se constituer, le ralliement de ces acteurs dominants contribue, par la crédibilité qu'ils apportent au nouveau réseau (dans un secteur où l'importance des anticipations n'est plus à démontrer), à sa consolidation et à son développement. Toutefois le caractère tardif de leur conversion fait que ces acteurs détiennent un pouvoir beaucoup plus faible sur le contrôle et l'évolution du nouveau réseau que celui qu'ils détenaient dans l'ancien. Au départ limité à des niches technologiques intéressant des minorités aux comportements, aux compétences et aux préférences différentes²⁹, le réseau peut ainsi se déployer jusqu'à atteindre une force équivalente au réseau déjà existant.

²⁷ Certaines innovations techniques permettent de franchir "un obstacle épistémologique (...) à l'aide d'un alliage inédit, dont les éléments sont aussi bien matériels qu'immatériels, autrement dit des savoir-faire techniques comme des connaissances scientifiques" (Emmanuel Saint-James, 1993, p. 3).

²⁸ Par exemple, on a pu observer cette myopie face aux évolutions à l'œuvre dans les comportements d'IBM lors de l'apparition de la micro-informatique, ou de Microsoft au début du développement d'Internet.

²⁹ Jean-Michel Dalle (1995) à partir du modèle de standardisation technologique de Brian Arthur, auquel il intègre une dimension spatiale à côté de sa dimension temporelle, démontre la possibilité que plusieurs standards technologiques cohabitent. La situation de l'informatique dans son ensemble (coexistence de l'informatique traditionnelle, du monde Unix et de la micro-informatique), comme dans certains de ses sous-secteurs (coexistence du monde compatible PC et du monde Apple), vérifie cette théorie.

C'est ce cadre d'analyse qui est utilisé pour étudier la constitution successive des réseaux technico-économiques dans l'informatique (résumée dans le tableau IV) ³⁰.

Section II - L'histoire de l'informatique : une histoire de réseaux

Schématiquement, on peut considérer que chaque décennie a vu l'émergence d'un nouveau réseau technico-économique dans l'informatique, la situation actuelle voyant la fusion tendancielle de ces différents réseaux (section III). Bien que certains de ces réseaux apparaissent comme fortement contrôlés par un acteur dominant, celui-ci n'a pu empêcher la constitution de nouveaux réseaux, confirmant dans le cas de l'informatique l'affirmation de Bruno Bellon selon laquelle "les avantages liés aux économies d'échelle, à l'apprentissage, ou aux autres barrières à l'entrée – atouts absolus en statique – sont toujours remis en cause, ne serait-ce qu'avec le déplacement des marchés" (1991, p. 491). Toutefois si chacun de ces réseaux correspond à l'apparition de nouveaux types d'ordinateurs (*mainframes*, mini-ordinateurs et stations de travail, micro-ordinateurs), il importe de ne pas analyser la diffusion des générations successives d'ordinateurs comme s'il s'agissait uniquement de produits distincts : en effet, "les perspectives d'adoption de chacune des nouvelles générations sont fortement influencées par le nombre d'ordinateurs des générations précédentes déjà installés et par les programmes utilisés" (Chris Freeman, 1988, p. 44).

A - LES ANNEES 60 ET LA CONSTITUTION D'UN PREMIER RESEAU : L'INFORMATIQUE TRADITIONNELLE.

Au début des années soixante, la production d'ordinateurs est très largement dominée par IBM³¹, qui bénéficie des avantages des *first movers* (Alfred D. Chandler, 1992, p. 70) avec une faible part du marché pour les autres constructeurs (les "sept nains"). Il n'existe aucune standardisation au niveau du matériel, pas même au niveau des différents ordinateurs

³⁰ Si cette analyse permet d'expliquer l'évolution de la grande majorité de l'activité informatique, elle n'intègre pas certains segments très spécifiques comme les supercalculateurs.

³¹ Si le premier ordinateur commercialisé l'a été par Remington Rand, IBM s'imposera rapidement en raison de ses compétences préexistantes issues de son activité dans les systèmes de cartes perforées (Laurent Le Dortz, 1997, p. 255-256). Dans les années cinquante, IBM consolidera sa domination par la pratique de la location des ordinateurs (qui constitue une barrière à l'entrée efficace), refusant même de vendre sa production jusqu'à ce qu'une décision de justice en 1956 ne la contraigne à le faire (Jean-Pierre Brulé, 1993, p. 89).

produits par IBM qui sont regroupés dans cinq gammes incompatibles entre elles. La seule standardisation existante se situe au niveau des langages de programmation (Fortran pour les applications scientifiques et Cobol pour les applications de gestion) chaque constructeur développant des compilateurs pour traduire les programmes écrits dans ces langages en instructions exécutables sur leurs matériels³². Les difficultés pour faire émerger de nouveaux langages de programmation sont déjà riches d'enseignements sur les processus de standardisation. Si un acteur dominant peut empêcher l'émergence d'un standard, il lui est plus difficile d'imposer un nouveau standard rompant radicalement avec le standard précédent : la *Conférence on Data System Languages (CODASYL)* impulsée par le Ministère de la Défense des Etats-Unis, qui regroupait l'ensemble des acteurs de l'informatique et qui avait créé Cobol en 1960, n'arrivera pas à imposer un langage pourtant prometteur (Algol) en raison de l'opposition de fait d'IBM ; par contre la toute puissance d'IBM n'est pas suffisante pour que son projet de langage universel (PL1) soit adopté massivement.

La constitution du premier réseau technico-économique³³ déterminant dans l'informatique date de la décision d'IBM de produire une série d'ordinateurs (les 360) compatibles entre eux et devant répondre à l'ensemble des besoins aussi bien scientifiques que de gestion (Christian Genthon, 1995, p. 41). La simple annonce de cette décision effectuée simultanément le 7 avril 1964 dans 62 villes et 14 pays étrangers (Jean-Michel Treille, 1973, p. 16) suffit à déstabiliser les entreprises concurrentes (RCA et Control Data baissent leurs prix de 30 %) en développant l'attentisme parmi la clientèle. Cette opération, dont le coût est estimé par IBM à 5 milliards de dollars³⁴, durera deux ans avant que les premiers ordinateurs IBM 360 soient introduits sur le marché et assurera à IBM "le contrôle définitif³⁵ du marché mondial des ordinateurs" (Jean-Michel Treille, 1973, p. 44). Elle introduit la notion même de standard au cœur de l'industrie informatique en effectuant une distinction conceptuelle entre

³² La standardisation n'est que relative, les compilateurs produits par les constructeurs intégrant fréquemment de nouvelles instructions au langage existant et générant dans certains cas des comportements spécifiques, qui rendaient difficiles le portage d'une application sur une autre plate-forme matérielle.

³³ En considérant que la constitution du réseau technico-économique qui a permis aux ordinateurs numériques de s'imposer rapidement dans les années cinquante par rapport aux ordinateurs analogiques, relève de la "préhistoire de l'informatique".

³⁴ Cette estimation, qui représente le double du financement de la première bombe atomique, a été vraisemblablement exagérée par IBM dans le but de dissuader la concurrence (Jean-Michel Treille, 1973, p. 42).

³⁵ Ce contrôle était très largement perçu comme "définitif" en 1973. La suite, difficilement prévisible montrera que ce n'était pas tout à fait exact.

architecture et structure des ordinateurs : l'architecture est constituée par l'ensemble des règles et des conventions d'utilisation, dont par exemple la représentation des informations, le mode d'adressage, le jeu d'instructions³⁶, alors que la structure de chaque machine concerne les caractéristiques d'une réalisation particulière de l'architecture (Gérard Dréan, 1996 A). La création de ce nouvel intermédiaire (une architecture standardisée) par IBM, permettait de "traduire" les attentes des utilisateurs (principalement des très grandes entreprises et des administrations, notamment l'armée) qui "pouvaient évoluer dans la gamme et profiter des progrès technologiques sans remettre en cause leurs investissements passés" et étaient enfin réunis "dans une même communauté, facilitant les échanges entre eux et accélérant la diffusion des innovations" (idem, p. 22). Mais en même temps, elle allait permettre d'englober dans ce réseau des producteurs de matériel informatique autres qu'IBM. "En séparant les notions d'architecture et d'implémentation, et en introduisant la notion d'interface standard entre unités, IBM ouvre involontairement la voie aux offreurs de compatibles" (idem, p. 20). Cette production de "matériels compatibles" s'effectue dans deux directions : tout d'abord au niveau des périphériques (imprimantes, lecteurs de bandes et disques, mémoires...), il devient possible pour d'autres producteurs "d'offrir des produits aisément substituables aux produits d'IBM, en ayant des prestations supérieures et/ou des prix inférieurs" (Christian Genthon, 1995, p. 48) ; ensuite au niveau de l'unité centrale (le cœur de l'ordinateur), la stratégie de production d'ordinateurs "compatibles", initiée avec une réussite inattendue par Gene Amdahl³⁷, fut suivie par un certain nombre de constructeurs d'ordinateurs pour tout ou partie de leur production (Magnuson, Intel, NAS, RCA, Bull, Hitachi...). La production de matériels compatibles va contribuer à étendre le réseau constitué autour d'IBM au détriment des constructeurs non compatibles, en permettant aux utilisateurs de bénéficier de l'environnement IBM avec une plus grande liberté de choix et des produits plus compétitifs en terme de prix et de performances (Michel Delapierre, Jean-Benoît Zimmermann, 1991, p. 51)³⁸. Le réseau va également s'étendre à de nombreuses sociétés de logiciels et de services, suite à la décision d'IBM en 1969, sous la pression des procédures judiciaires en cours,

³⁶ Le jeu d'instructions du système 360 détient à ce jour un record absolu de longévité (Jacques Printz, 1998, p. 235).

³⁷ Quand Gene Amdahl quitte IBM en 1967 pour produire des ordinateurs compatibles, IBM détient entre 60 et 70% du marché et il paraît inconcevable qu'une entreprise nouvelle puisse produire des machines moins chères, plus puissantes, et surtout persuader des clients de l'acheter (Godefroy Dang Nguyen, 1995, p. 113).

³⁸ Le même phénomène se reproduira dans la microinformatique avec la fabrication de "clones" de l'IBM PC dont les premières victimes seront les fabricants de micro-ordinateurs non compatibles.

d'effectuer une tarification séparée du logiciel et des services (*unbundling*). Si ce réseau comprend en proportion croissante des producteurs (d'ordinateurs, de périphériques, de logiciels, de services) autres qu'IBM, il reste largement contrôlé par ce dernier. IBM pratique une "stratégie d'accommodement dynamique" (Godefroy Dang Nguyen, 1995, p. 113) en maintenant des prix et des marges élevés et en laissant ses concurrents pénétrer graduellement le marché, tout en conservant sur eux, par sa capacité à faire évoluer les standards de fait, un pouvoir considérable³⁹.

L'utilisation principale de l'informatique était le calcul scientifique et l'informatisation, dans les grandes entreprises et les administrations, des tâches de *back office* : gestion administrative, gestion du personnel, des stocks, gestion financière... L'informatique s'introduisait dans des domaines "où la rapidité du traitement s'accordait parfaitement avec la contrainte de travaux répétitifs sur de gros volumes d'informations" (Jean-Michel Treille, 1973, p. 186), "son vrai triomphe étant d'abaisser chaque année le coût d'une ligne de facture" (Jean-Pierre Brule, 1993, p. 47). Si elle permettait d'étendre l'automatisation aux activités de bureaux, elle ne faisait qu'automatiser les tâches d'un ensemble borné de fonctions (Alain Rallet, 1997, p. 96-98). Cette "informatique lourde de gestion" n'introduisait pas de véritable rupture avec la mécanographie dont elle était issue (François Pichault, 1990).

L'ensemble de ce réseau partageait une même vision d'une informatique très centralisée et élitiste, dont l'invention du terme ordinateur, pour désigner la mise en ordre automatique de l'information par un acteur unique, est significative. C'est l'époque où le pouvoir des informaticiens est à son apogée, l'utilisateur novice se trouvant contraint de s'adapter à cet objet quasi mythique qu'est l'ordinateur : l'informatique est entre les mains d'un service de spécialistes (le département informatique) qui développent les programmes et répondent à des demandes d'utilisateurs qui consultent en retour des listings (Alain Rallet, 1997, p. 97).

C'est cette cécité face à une autre conception possible de l'informatique, utilisable par la majorité des entreprises et de leurs services avec la mini informatique, puis à toutes les entreprises et à une proportion croissante de ménages avec la microinformatique, qui

³⁹ Par exemple RCA engagé avec un certain succès dans une politique de compatibilité maximale avec les 360 d'IBM vit sa part de marché chuter brutalement avant de disparaître quand IBM modifiera certaines caractéristiques des 360 (Jean-Pierre Brule, 1993, p. 133).

reléguera IBM au rang "d'un acteur parmi d'autres, participant à un jeu concurrentiel dont elle ne fixe plus les règles" (Gérard Dréan, 1996 A, p. 45).

B - LES ANNEES 70 ET LA CONSTITUTION D'UN DEUXIEME RESEAU BASE SUR LA MINI INFORMATIQUE ET LES STATIONS DE TRAVAIL.

La mini informatique fut fondée par des anciens salariés de l'informatique "traditionnelle" et notamment d'IBM (Ken Olsen pour DEC, An Wang pour Wang). Le mini-ordinateur, inventé par DEC (PDP 8), est rapidement produit par de nombreuses entreprises, nouvelles⁴⁰ ou anciennes. Si la mini informatique ne représente pas une véritable rupture technique (ces machines ne mettent pas en œuvre une technologie ou des principes différents de ceux des ordinateurs traditionnels), le mini-ordinateur se caractérise par un moindre encombrement, une plus grande convivialité, et un coût beaucoup moins élevé (Jean-Marie Desaintquentin et Bernard Sauter, 1991, p. 14). Ce mouvement se prolonge avec l'apparition des stations de travail, autour des processeurs RISC, produites par les mêmes constructeurs et par de nouveaux acteurs, dont la réussite la plus spectaculaire est Sun⁴¹. Le nombre d'ordinateurs installés croît rapidement pour atteindre 300 000 aux Etats-Unis en 1976 (Frédéric Dromby, 1999, p. 548).

Ces ordinateurs vont être installés dans les différents services des grandes entreprises et des administrations (autorisant une informatique moins centralisée) et dans les moyennes entreprises. Ils vont permettre une utilisation plus diversifiée et l'informatisation de procédés plus complexes (conception, modélisation), notamment dans l'industrie où ils servent de support à des applications de Conception Assistée par Ordinateur et de Gestion de la Production Assistée par Ordinateur (Claire Charbit, Jean-Benoît Zimmermann, 1997, p. 12).

Cette informatique se distingue par l'absence d'un acteur dominant. L'intermédiaire qui permettra à ce réseau de se constituer est le système d'exploitation Unix. Unix fut développé au début des années soixante dix par des chercheurs des Bell Labs, K. Thompson et D. Ritchie (rejoints par Brian Kernighan), en adaptant aux mini-ordinateurs (initialement le PDP-11 de

⁴⁰ Michel Delapierre et Jean-Benoît Zimmermann (1984, p. 12) dénombrent 140 nouvelles créations d'entreprises sur le seul segment des mini-ordinateurs aux Etats-Unis de 1977 à 1982.

⁴¹ SUN (Stanford University Network) Microsystems est née en 1982 et est devenue le leader des stations de travail en 1988 en supplantant Apollo Computer, l'inventeur de ces machines.

Digital), le système Multics, système d'exploitation le plus avancé de l'époque, multitâches et multiutilisateurs. Ce système, resté inachevé – il avait été abandonné faute de ressources en 1968 – était destiné aux gros ordinateurs de General Electric et était le fruit d'une collaboration de cette entreprise avec le MIT et les Bell Labs, soutenu par des financements fédéraux. ATT ne pouvant pas commercialiser Unix en raison de sa position monopolistique dans les télécommunications⁴², la licence d'Unix fut proposée à des conditions très libérales à l'Université de Berkeley : celle-ci disposait du code-source et de la possibilité de le modifier et de le redistribuer, le système originel restant cependant propriété d'ATT. L'Université de Berkeley va développer une version d'Unix appelée BSD, vendue environ 1000 \$ avec possibilité de réaliser des copies des exécutable et de modifier le code source. Elle sera d'abord diffusée dans le monde universitaire, puis plus largement grâce au développement des stations de travail. En particulier Unix sera choisi par Sun pour se positionner sur ce marché, où elle obtiendra un large succès notamment grâce à l'écriture de NFS (Network File System), un logiciel qui permet des échanges de fichiers entre stations de travail de constructeurs différents.

Le caractère ouvert d'Unix, indépendant d'un constructeur, son écriture dans un langage de haut niveau (le langage C développé en parallèle par D. Ritchie et B. Kernighan, qui rend Unix assez facilement portable sur différents matériels) lui permettra de bénéficier d'une forte dynamique d'innovation (notamment de la part de la communauté universitaire) et d'être implémenté par la plupart des constructeurs de mini-ordinateurs et de stations de travail. En conséquence de son démantèlement en 1984, ATT peut avoir une activité informatique marchande, et, étant propriétaire de droit d'UNIX (du système et de son nom), a la possibilité de demander des redevances importantes aux différents utilisateurs. Chaque constructeur s'oriente alors vers le développement d'un système d'exploitation propre constitué d'une combinaison d'éléments standards et d'éléments originaux, ce que facilite l'architecture d'Unix conçu de façon originale comme un assemblage de fichiers, de modules ayant chacun une fonction particulière (Nicolas Jullien, 1999, p. 48) : à côté de la version d'ATT (UNIX System V), existent BSD de l'Université de Berkeley, Solaris de Sun, AIX d'IBM, HP-UX d'Hewlett

⁴² Le *Consent Decree* de 1956 soumettait ATT à une stricte limitation de ses activités au domaine réglementé du téléphone, et lui interdisait toute activité marchande dans des domaines tels que les composants ou l'informatique.

Packard, Digital UNIX de DEC, SCO sous licence Microsoft⁴³. Chaque système possède des caractéristiques particulières avec apparition d'incompatibilités entre les différents systèmes, ce qui a comme conséquence économique la coexistence de plusieurs marchés captifs (Nicolas Jullien, 1999, p. 51) et donc limite les économies d'échelle possibles.

Les tentatives d'accord sur un standard commun ont été aussi nombreuses qu'infructueuses. En 1987, ATT passe un accord avec Sun pour faire converger Unix System V d'ATT et SUN OS (Unix Berkeley 4.3) qui suscite une réponse des concurrents sous la forme de l'Open Software Foundation en 1988 (Apollo, Bull, Digital, HP, IBM, Nixdorf, Philips et Siemens), puis la création du groupe X/Open par les constructeurs européens bientôt rejoints par les constructeurs américains et par de grands utilisateurs. En juin 1993, Novell rachète Unix System Laboratories à ATT. Par la suite Novell va permettre l'utilisation libre de la marque, dont elle souhaite confier la gestion au consortium X/Open : toute entreprise pourrait utiliser librement le nom d'Unix à condition de respecter les spécifications du standard X/Open. Mais IBM, HP, SUN et SCO estimèrent que Novell cherchait en fait à imposer *de facto* sa propre version d'Unix, UnixWare ce qui empêcha de fait ce projet d'aboutir (Carl Shapiro, Hal R. Varian, 1999, p. 226).

L'existence de systèmes "propriétaires" concurrents en perpétuelle évolution a certainement entravé le développement de ce réseau⁴⁴ qui est pourtant "un foyer d'innovations privilégié" (Gérard Dréan, 1996 A, p. 56) pour toute l'informatique (notamment dans le multiprocessing, les systèmes distribués, la disponibilité et la sécurité). On peut dire que ce réseau s'est structuré autour d'une traduction imparfaite qui a nui à son extension. Curieusement c'est peut-être la naissance d'un système d'exploitation Unix libre, conçu à l'origine pour micro-ordinateur (Linux) et qui est progressivement implémenté sur l'ensemble des machines Unix, qui va faire évoluer favorablement cette situation (Nicolas Jullien, 1999).

Par contre, la constitution de ce deuxième réseau correspond à un déclin relatif d'IBM, qui va mettre du temps avant de comprendre toutes les potentialités de cette nouvelle informatique. Il est vrai qu'initialement T.J. Watson Senior, PDG d'IBM estimait que la diffusion des ordinateurs serait limitée à une poignée de clients (Chris Freeman, 1988, p. 44).

⁴³ En 1985, il existe plus de 100 versions différentes (Pamela Gray, 1993, p. 95).

⁴⁴ Par exemple, Informix producteurs de logiciels applicatifs (notamment des bases de données) doit adapter ses produits à 305 systèmes différents de 72 constructeurs de machines (Pamela Gray, 1993, p. 109).

Cette situation d'inertie de la firme dominante a été étudiée par J.L. Bower et C.M. Christensen (1995), dans le cas de l'industrie des disques durs pour ordinateur, en termes d'innovation " disruptive ". La principale innovation dans l'industrie des disques durs a été l'introduction d'un disque de format toujours plus réduit (14 pouces, puis 8, puis 5,25 et enfin 3,5). La caractérisation d'innovation " disruptive " provient du fait que les progrès en termes de poids et d'encombrement, s'accompagnent au début d'une moindre performance sur le point crucial de la capacité de stockage. De ce fait, cette innovation est délaissée par l'entreprise leader sur les disques du format précédent car ces principaux clients demandent des disques de grande capacité, et ce sont des nouveaux entrants qui à partir de marchés émergents (PC, portables) introduisent l'innovation de format. Le comblement rapide du handicap de capacité des nouveaux disques amène ces firmes auparavant marginales à supplanter les anciens leaders du marché, jusqu'à ce qu'un nouveau format apparaisse. Cette analyse peut être étendue au cas plus général des ordinateurs, dont les nouvelles générations (minis, puis micros) se caractérisent par des coûts beaucoup plus faibles mais par une capacité de traitement moindre. En termes de réseaux technico-économiques, on peut dire que ce sont de nouvelles firmes, qui ne sont pas " prisonnières " de l'ancien réseau technico-économique, qui peuvent traduire en de nouveaux produits les besoins latents et non exprimés de nouveaux utilisateurs, correspondant à de nouveaux usages, et contribuer à la création d'un nouveau réseau technico-économique, qui va s'étendre progressivement au détriment de l'ancien réseau. C'est ce qui se passera à nouveau avec l'irruption de la micro-informatique dans les années quatre-vingt.

C - LES ANNEES 80 ET LA CONSTITUTION D'UN TROISIEME RESEAU AUTOUR DE LA MICROINFORMATIQUE.

La microinformatique va introduire une rupture radicale avec l'informatique traditionnelle. Elle s'appuie certes sur une innovation technique (le microprocesseur avec l'invention du 4004 par Intel en 1971) mais elle n'a pu se développer que parce qu'elle entrait en résonance avec de nouvelles aspirations culturelles.

L'absence d'IBM au début de la microinformatique, qui disposait pourtant des connaissances techniques nécessaires dès la fin des années 60, montre l'importance de ce facteur. La vision d'IBM de l'informatique future restait celle de grands systèmes où la centralisation des informations était le principal moyen de les rendre contrôlables par la direction des entreprises, à l'image d'ailleurs de ce qui se passait à l'intérieur même d'IBM. De

même, David Ahl avait au sein de Digital conçu dès 1974 un Personal Computer mais sa proposition avait été rejetée par les ingénieurs des ventes qui ne voyaient pas à quel usage concret pourrait correspondre un ordinateur d'aussi faible capacité, comparé aux mini-ordinateurs développés par Digital (Philippe Breton, 1987, p 210-211). L'influence d'un contexte socio-culturel propice peut être vérifié a contrario par le fait que le premier ordinateur commercialisé l'a été en France (R2E équipé d'un microprocesseur Intel 8008 par Micral en 1975) mais sans connaître une grande diffusion.

Le milieu qui va littéralement porter les débuts de la microinformatique est celui de la contre-culture, de *l'underground* américain (Philippe Breton, 1990, p. 52), dont le cœur sont des étudiants de Berkeley en Californie, férus d'informatique, opposants à la guerre du Vietnam, et adversaires d'une informatique centralisée dont le symbole est le "gros" ordinateur qui met en fiche les citoyens. Cette conception d'une informatique dont les multiples possibilités doivent profiter aux individus, et pas seulement à l'Etat ou aux entreprises, est notamment développée par une association *The People's Computer Company*, fondée en Californie et qui bénéficie de l'aide et de l'implication de chercheurs du Stanford Research Institute et du Palo Alto Research Center de Xerox.

En même temps, ce besoin d'appropriation individuelle de l'informatique et de ses possibilités, rencontre un écho favorable du côté des utilisateurs de l'informatique traditionnelle, désireux de trouver une réponse à des besoins individuels mal satisfaits, et d'échapper à la tutelle envahissante des services informatiques centraux. C'est dans ce contexte que va se constituer dans la *Silicon Valley* un "district industriel", mélange de concurrence et de coopération entre les pionniers de la microinformatique.

La première entreprise qui saura faire converger ces éléments disparates en un véritable réseau technico-économique, permettant l'éclosion de la microinformatique, sera Apple. Les aspects déterminants de ses ordinateurs sont la convivialité et l'autonomie qu'ils apportent aux utilisateurs, notamment grâce à l'utilisation du tableur (VisiCalc développé en 1979 par Software Arts), produit qui a été inventé pour la microinformatique, à la différence des autres applications (traitement de texte, systèmes de gestion de bases de données...), qui existaient déjà sur les grands systèmes ou les mini-ordinateurs et qui seront simplement adaptés aux

micro-ordinateurs. Ce réseau très fortement convergent⁴⁵ est marqué par un certain élitisme, mais de nature plus sociale que technique, en s'adressant prioritairement aux professions libérales, aux cadres de direction et aux travailleurs du savoir et de l'édition graphique. Si ce positionnement a favorisé le décollage d'Apple, il constitue un obstacle à l'extension de ce réseau à l'ensemble des utilisateurs potentiels.

Apple va se retrouver positionné sur une "niche" de clients fidèles⁴⁶, mais quand les "marchés sont dominés par de fortes externalités de réseaux, les stratégies de niches sont intrinsèquement dangereuses" (Carl Shapiro, Hal R. Varian, 1999, p. 218). Selon ces auteurs c'est uniquement grâce aux prouesses impressionnantes du Macintosh et à la lenteur technologique de Microsoft, qu'Apple a pu survivre aussi longtemps avec cette stratégie de niche. Quand les autres micro-ordinateurs (compatibles PC) vont progressivement intégrer les innovations matérielles et logicielles d'Apple (écran graphique, icônes, utilisation de la souris) issues pour l'essentiel des travaux du Palo Alto Research Center de Xerox, la perte de spécificité du Macintosh face à des produits concurrents moins chers, va se traduire par une chute de sa part de marché avec l'augmentation du nombre d'utilisateurs, particulièrement nette au moment de la sortie de Windows 95 : la part de marché mondiale en micro-informatique d'Apple va passer de 9,4 % en 1993 à 5,3 % en 1996, puis à 3,3 % en 1997 (source IDC). Actuellement Apple en est réduit à affirmer son originalité dans le *design* de ses machines, avec l'iMac, qui s'il a été un incontestable succès en permettant à Apple de renouer avec les bénéfices⁴⁷, n'a pas fait remonter significativement sa part de marché.

Le réseau principal, qui va assurer le triomphe de la microinformatique, se crée à partir du ralliement d'IBM en 1981. Comprenant le décalage entre les exigences de la microinformatique et son propre modèle d'organisation adapté à l'informatique traditionnelle,

⁴⁵ "La notion de convergence est destinée à saisir le degré d'accord engendré par une série de traductions, et par les intermédiaires de toutes sortes qui les opèrent" (Michel Callon, 1991, p. 211).

⁴⁶ Cette stratégie se retrouve dans le choix d'une architecture "fermée" empêchant le développement de "clones" qui peuvent permettre le développement d'un réseau (cf. infra). Après s'être vivement opposé à l'existence de "clones", Apple changera d'attitude en 1996, la marginalisation progressive des micro-ordinateurs de type Mac ayant comme conséquence que de plus en plus d'éditeurs de logiciels ne développaient plus de version "Mac" de leurs produits. Mais ce développement tardif des "cloneurs" au lieu d'étendre le "monde Mac" s'est surtout effectué au détriment d'Apple et a pesé sur ses prix, contraignant Apple à racheter en septembre 1997 son principal "cloneur", Power Computing, pour tenter de récupérer le savoir-faire en matière de distribution *on-line* de cette société constituée par des anciens de Dell (Eurostaf, 1997 A, p.71).

⁴⁷ Alors qu'Apple avait perdu plus de 1,8 milliards de dollars en 1996 et 1997.

IBM va développer le PC selon un modèle *adhocratique* en laissant une liberté totale à une petite équipe isolée de la ligne hiérarchique (Godefroy Dang Nguyen, 1995, p. 174). Celle-ci, tirant les leçons de l'échec d'IBM dans la mini informatique à cause de sa lenteur à réagir, va accélérer le développement en sous-traitant la plupart des composants du PC (notamment le microprocesseur à Intel et le système d'exploitation à Microsoft)⁴⁸, en rupture avec la tradition d'IBM⁴⁹. Le succès est énorme, la puissance d'IBM fait converger les anticipations⁵⁰ et de multiples acteurs s'intègrent au réseau constitué par les standards *de facto* de l'IBM PC : par la fabrication de "clones", le ralliement de tous les constructeurs de micro-ordinateurs (sauf Apple) à la production d'ordinateurs compatibles, la naissance de nombreux éditeurs de progiciels très divers pour les PC. S'enclenche à un rythme très rapide un cercle vertueux constitué d'économies d'échelle, de baisse des prix et d'amélioration des performances, dans des structures plus éclatées et, pour certaines, très concurrentielles⁵¹, et d'extension du nombre d'utilisateurs.

L'utilisation de l'informatique s'étend à des domaines nouveaux (outils de productivité personnelle, aide à la décision, créativité personnelle). L'automatisation et la rationalisation de la gestion de l'information au sein de l'entreprise, s'étendent au niveau même des postes de travail (Eric Brousseau, 1993, p. 176). Cette diffusion des applications informatiques à l'ensemble des fonctions et des postes de travail de la firme s'opère grâce à une technologie

⁴⁸ L'IBM-PC est "contrairement à toutes les autres machines d'IBM, un système délibérément ouvert aux logiciels et aux extensions d'autres fournisseurs". "IBM se pose comme catalyseur d'une galaxie d'entreprises indépendantes, en comptant sur leur créativité et leur dynamisme pour faire de son matériel le standard de fait" (Gérard Dréan, 1996 A, p. 47-49).

⁴⁹ Sur un coût de production de 860 dollars de l'IBM PC en 1985, 625 dollars sont des fabrications extérieures à IBM (Christian Genthon, 1995, p. 110).

⁵⁰ Les différents acteurs supposaient que IBM aurait le même poids dans la microinformatique que celui qu'elle avait dans l'informatique traditionnelle. Même si cette prévision était erronée, le fait qu'elle soit largement partagée suffit à assurer le succès des choix technologiques de l'IBM PC par le développement de multiples progiciels et matériels "compatibles PC".

⁵¹ A l'opposé du modèle généraliste intégré de l'informatique traditionnelle où chaque offreur produit la totalité du système informatique, la microinformatique (à l'exception notable de Apple) se caractérise par une spécialisation des entreprises sur un segment particulier (microprocesseurs, autres composants matériels, assemblage de micro-ordinateurs, système d'application, progiciels d'applications...). En conséquence de la standardisation réalisée sur l'architecture de l'IBM PC, "la concurrence se joue entre les composants et non plus entre les systèmes" (Carl Shapiro, Hal R. Varian, 1999, p. 206). Cette structure, comprenant des secteurs concentrés et d'autres dispersés, va se révéler particulièrement performante provoquant l'inversion de la "loi de Gresham" qui énonçait qu'à coût égal un grand système informatique offrait plus de puissance que plusieurs petits systèmes : à l'heure actuelle, le prix du MIPS (million d'instructions par seconde) sur un PC est estimé à un millième de ce qu'il est sur un grand système (Gérard Dréan, 1996 A, p. 297).

centrée sur le développement d'interfaces avec les usagers et donc plus facilement "domesticable" (Alain Rallet, 1997, p. 97). Ce développement rapide de la microinformatique au détriment de l'informatique traditionnelle et de la mini informatique (*downsizing*) s'accompagne d'une perte de pouvoir des départements informatiques au sein des entreprises et des administrations.

Deux phénomènes dans l'histoire du succès du micro-ordinateur PC confirment certaines conclusions des théories de la diffusion et de la compétition technologique.

Le premier est l'importance décisive que peuvent prendre des "petits événements", en l'occurrence la décision d'IBM de confier, dans des conditions assez rocambolesques, l'écriture du système d'exploitation, à Microsoft⁵². Microsoft est à l'époque un petit producteur de logiciels qui emploie 50 personnes et qui avait été fondé en 1975 pour commercialiser un interpréteur de Basic écrit pour les premiers micro-ordinateurs Altair. Microsoft qui, jusqu'à cette époque ne commercialise que des langages de programmation, va développer MSDOS à partir du rachat pour 50 000 dollars du système Q-DOS (dont les initiales signifient *Quick and Dirty Operating System* !) à la petite société Seattle Computer (Roberto Di Cosmo et Dominique Nora, 1998, p. 31-32). Le choix de Microsoft, qui à l'époque apparaît anodin, va permettre à Microsoft de devenir l'acteur dominant de la microinformatique par sa maîtrise de l'évolution du standard principal (le système d'exploitation), en alliance avec Intel qui détermine l'évolution des microprocesseurs. Pour IBM, le choix d'une architecture délibérément ouverte à d'autres entreprises était destiné à occuper rapidement le marché de la micro-informatique et devait rester provisoire dans un secteur jugé assez marginal. La très forte croissance de la micro-informatique incitera IBM, qui occupe une place minoritaire dans un secteur qu'elle a paradoxalement très largement contribué à créer, à essayer d'en reprendre le contrôle en tentant d'imposer un nouveau système d'exploitation (OS/2 en 1988), mais ce sera un échec malgré l'importance des efforts consentis⁵³ : "trop de systèmes et d'extensions

⁵² Au départ IBM avait contacté Gary Kildall dirigeant de Digital Research qui produisait le principal système d'exploitation pour micro-ordinateur, CP/M. Le peu d'empressement de celui-ci et surtout son refus des contraintes juridiques de "non-divulgation" posées en préalable par IBM, conduisirent IBM à se tourner vers Microsoft (Robert X. Cringely, 1994, p. 119-120).

⁵³ Le développement d'OS/2 a été estimé à deux milliards de \$ et a mobilisé le travail de plusieurs années d'un millier des meilleurs programmeurs d'IBM.

compatibles avec le PC ont été commercialisés, trop de logiciels ont été écrits pour qu'un retour en arrière soit possible."(Gérard Dréan, 1996 A, p. 49)⁵⁴.

Le deuxième phénomène est que cette situation de verrouillage technologique s'est opérée sur une solution non optimale, le système d'exploitation de Microsoft (MSDOS) étant peu performant sur les plans de la technique et de la convivialité⁵⁵. Microsoft va transformer les faiblesses de départ de son système d'exploitation en avantage commercial⁵⁶, en mettant régulièrement sur le marché de nouvelles versions de son système d'exploitation, puis en développant une interface graphique (Windows) pour combler son manque de convivialité, enfin en créant Windows 95 dont tout le monde savait avant même sa sortie, et quelles que soient ses performances⁵⁷, qu'il constituerait le nouveau standard de la micro-informatique⁵⁸. Cette maîtrise par Microsoft de l'évolution de la micro-informatique (sortie en fonction de ses intérêts des nouvelles versions de son système d'exploitation dont elle était seule à connaître les spécifications) et les retombées financières qui en résultent⁵⁹ lui ont donné un avantage décisif sur ses concurrents dans la production des différents progiciels (traitement de texte, tableur, système de gestion de bases de données...) où Microsoft est devenu hégémonique avec près de 90 % du marché⁶⁰. La puissance acquise par Microsoft et l'élargissement du champ des rendements croissants d'adoption lui permet d'essayer d'étendre son emprise au

⁵⁴ La sanction pour IBM de la succession de ces erreurs sur l'évolution de l'informatique est sévère : en moins de dix ans (1985-1993), IBM va passer du statut de société réputée pour la fidélité de ses clients et l'importance de ses bénéfices, à celui d'entreprise réputée pour l'énormité de ses pertes : 2,8 milliards de dollars en 1991, 4,96 milliards de dollars en 1992, et enfin le record de 8,1 milliards de dollars en 1993.

⁵⁵ Un autre système d'exploitation (celui de Digital Research) était généralement jugé plus performant, mais Microsoft a utilisé sa position de producteur le plus puissant dans les langages de programmation pour l'évincer. Microsoft n'a sorti qu'avec beaucoup de retard des langages pour ce système, qui ont été commercialisés 50 % plus chers que les mêmes langages pour le MS-DOS (Michael Cusumano, Richard Selby, 1996).

⁵⁶ De façon plus générale Roberto Di Cosmo et Dominique Nora estiment que Microsoft a une "remarquable aptitude à transformer des échecs techniques en succès commerciaux" (, 1998, p. 34).

⁵⁷ Carl Shapiro et Hal R. Varian estiment que "Windows 95 est un "machin" pour rester poli, rafistolé de tous les côtés pour permettre aux programmes tournant sous DOS de continuer à fonctionner, de même qu'aux programmes conçus pour les anciennes versions de Windows" (1999, p. 244).

⁵⁸ Plus d'un million d'exemplaires ont été vendus les trois premiers jours.

⁵⁹ Microsoft a réalisé en 1997 un bénéfice de 3,5 milliards de dollars pour un chiffre d'affaires de 11,4 milliards de dollars.

⁶⁰ L'étude des problèmes de qualité des logiciels et des différentes pratiques pour les imposer sur le marché est développée ultérieurement (cf. notamment chapitre VII).

reste de l'informatique (avec notamment Windows NT pour les serveurs) voire à l'ensemble des technologies numériques.

Section III - Situation actuelle et perspectives

A - LA SITUATION ACTUELLE : LA FUSION TENDANCIELLE EN UN SEUL RESEAU.

L'importance quantitative de chacun de ces trois réseaux est difficile à apprécier. Sur le plan du matériel, les frontières entre mainframes, mini-ordinateurs, stations de travail et micro-ordinateurs ne sont pas étanches et se déplacent en fonction des évolutions techniques et économiques. En terme de parc installé, Jean-Marie Desaintquentin et Bernard Sauter (1996, p. 19) donnent les ordres de grandeur suivants :

Tableau I

Type de matériel	Prix du matériel	Prix du logiciel	Parc mondial
Mainframe	5 000 000 F	500 000 F	50 000
Mini-ordinateur	500 000 F	50 000 F	500 000
Micro-ordinateur	50 000 F	5 000 F	1 800 000

En part d'un marché mondial du matériel informatique évalué à 148 milliards de dollars en 1996, la situation est la suivante :

Tableau II

Type de matériel	% du marché mondial
Grand système	22,0 %
Mini-ordinateur	17,6 %
Station de travail	14,0 %
Micro-ordinateur	46,5 %

Source : US Department of Commerce

Une appréciation plus pertinente nécessite d'intégrer les logiciels et les services – leur importance est supérieure à celle du matériel – mais est rendue difficile par l'absence de ventilation de l'activité de nombreuses entreprises. Gérard Dréan (1996 A, p. 57) fournit l'estimation suivante : en 1992, l'informatique traditionnelle représentait 45 % des ventes, le monde Unix 13 %⁶¹ et la micro-informatique 41 %.

En termes d'évolution, le contraste est saisissant entre l'évolution des PC et stations de travail et le reste de l'informatique (sur la période le taux de croissance annuel moyen de l'ensemble était de 5 %) :

Tableau III
Evolution des dépenses relatives au matériel informatique dans la zone OCDE

Type de matériel	Taux de croissance annuel moyen 1987 / 1994
Grands systèmes	-3,8 %
Moyens systèmes	-3,4 %
Petits systèmes	0,9 %
PC et stations de travail	10,9 %

Source : IDC et OCDE, in Eurostaf, 1996 B, p. 3

Le regroupement des PC et des stations de travail dans la même catégorie s'explique par le fait que les PC rattrapent les stations de travail en termes de puissance, avec des composants beaucoup moins chers en raison des économies d'échelle : "il se vend 100 fois plus de PC que de stations de travail (...). Il faudrait une nouvelle révolution architecturale comme le Risc par rapport au Cisc, pour que les stations de travail puissent se démarquer... mais on ne la voit pas venir" (Christian Genthon, Denis Phan, 1999, p. 183). Au contraire on assiste à l'émergence de "machines mixtes" PC/stations de travail ou stations de travail personnelles dont la croissance est très rapide (200 % en 1994, Eurostaf, 1995, p. 19). L'utilisation de plus en plus fréquente sur les stations de travail et les petits serveurs de

⁶¹ Il est vraisemblable que la part beaucoup plus grande de logiciels et de services non-marchands dans le monde Unix conduit à une sous-estimation de son importance réelle. Pour sa part, OpenGroup évalue le marché Unix à 39 milliards de dollars en 1996 et prévoit qu'il atteindra 50 milliards de dollars en 2000 (Eurostaf, 1997 A, p. 32).

systèmes d'exploitation écrits à l'origine pour des micro-ordinateurs (Windows NT et Linux) renforce cette convergence.

D'autre part, il n'est pas évident que le déclin des grands systèmes et des mini ordinateurs (*downsizing*) se poursuive. Selon Eurostaf ce processus s'est arrêté en 1994 (Eurostaf, 1995 A, p. 31-32), en raison de la baisse du prix du MIPS sur les grands ordinateurs⁶² et de la nouvelle utilisation des mainframes comme de puissants serveurs pour répondre aux besoins nés du développement des réseaux informatiques.

En effet, l'événement déterminant dans l'évolution actuelle de l'informatique est le développement des réseaux (au sens physique du terme) entre des ordinateurs de tous types. De plus en plus d'utilisateurs disposent d'un poste de travail multifonctionnel qui se substituent à l'utilisation de matériel dédié à une fonction spécifique. Ce poste de travail est le plus souvent un micro-ordinateur connecté à un réseau : en France, la part des salariés qui se servent d'un micro-ordinateur *relié* passe de 14 % en 1993 à 36 % en 1998 (Secrétariat d'Etat à l'Industrie, 1999, p. 67). De même Eurostaf (1996 A, p. 81) estime que la part des micro-ordinateurs connectés en réseau est passé de 45 % en 1993 à plus de 75 % en 1997. Une indication convergente est fournie par le fait que 63 % des entreprises industrielles disposent d'un réseau de micro-ordinateurs en 1997 alors qu'elles n'étaient que 32 % en 1994 ; dans le même temps, la proportion d'entreprises utilisant un système centralisé passe de 36% en 1994 à 39 % en 1997 (Secrétariat d'Etat à l'Industrie, 1999, p. 68). L'organisation du système informatique des grandes entreprises repose de plus en plus fréquemment sur trois niveaux : des gros ordinateurs qui stockent et traitent les données les plus importantes, des serveurs multiprocesseurs pour distribuer ces données, et des PC pour les consulter, le *Web* étant la toile de fond qui permet à ces trois niveaux de communiquer entre eux (Datamation, avril 1997, pp. 46-53).

En effet, la "connectivité généralisée des systèmes informatiques s'est réalisée grâce à un vecteur complètement inattendu, le réseau Internet" (Godefroy Dang Nguyen et Denis Phan, 1997, p. 189)⁶³. Internet peut être défini comme un réseau de réseaux interopérant via

⁶² Estimée à environ 30 % par an (Eurostaf, 1997 A, p. 32).

⁶³ Par exemple, en 1991, Jean-Marie Desaintquentin et Bernard Sauter, expliquait que TCP-IP, protocole de base d'Internet, "se présente comme un concurrent du modèle OSI qui, à terme, devrait pourtant le remplacer" (1991, p. 169).

des protocoles de transmission de l'information par paquets (*TCP Transmission Control Protocol*) et des protocoles de communication (*IP Internet Protocol*). Le succès de ces standards, confrontés à des normes publiques (les normes OSI de l'ISO) et à des standards "privés" déjà existants (SNA d'IBM, Netware de Novell, les protocoles d'AOL et de CompuServe, entreprises qui offraient des services en lignes sur des réseaux "privés" à des centaines de milliers d'utilisateurs⁶⁴...), s'explique par leur caractère ouvert, et leur développement par un organisme alternatif à la normalisation étatique et à la standardisation par le marché (François Horn, 1999 A)⁶⁵. Le développement de ces standards, effectué par la communauté universitaire, a été initialement financé par l'armée américaine⁶⁶, intéressée par le projet de constitution d'un réseau sans poste de pilotage central, où ce sont les serveurs eux-mêmes, et non le réseau, qui acheminent les données, et donc susceptible de résister à une attaque nucléaire. Cependant les principaux choix effectués - publication et simplicité des protocoles ce qui permettra leur implémentation gratuite sur tous les ordinateurs, ouverture du réseau avec la possibilité pour tout ordinateur ou pour tout réseau déjà existant de s'interconnecter, "sécurisation" faible jusqu'à une période récente des informations transmises - étaient assez éloignés des préoccupations militaires, et plus proches des besoins des chercheurs pour développer leurs travaux et leurs réflexions. Les standards à l'origine du World-Wide-Web (WWW), système d'information qui par sa simplicité d'utilisation a étendu l'utilisation de l'Internet à un large public non spécialisé⁶⁷ sont également des standards ouverts produits par un organisme alternatif. Ces standards (*HTML HyperText Markup*

⁶⁴ Par la suite AOL interconnectera son réseau à Internet et est devenu le plus important fournisseur d'accès à Internet avec plusieurs millions d'utilisateurs.

⁶⁵ Contradictoirement Carl Shapiro et Hal R. Varian estiment que pour des raisons d'efficacité, il est préférable de laisser l'établissement des standards aux forces du marché, ce que confirmerait l'expérience historique. Il est toutefois significatif que leur ouvrage consacré à l'économie de l'information qui traite abondamment des profondes transformations introduites par Internet et des processus de standardisation, ne consacre pas une seule ligne aux standards qui sont à la base d'Internet : les protocoles TCP-IP qui permettent les communications entre des machines hétérogènes ne sont même pas mentionnés ; quant à HTML qui est à la base du World-Wide-Web, il n'est évoqué qu'une seule fois pour simplement indiquer qu'il est une application du langage SGML, qui "n'a jamais vraiment pris car aucune entreprise suffisamment importante ne l'a sponsorisé" et qu' "il est en passe de courir le même danger de fragmentation qu'a connu Unix" (Carl Shapiro, Hal R. Varian, 1999, p. 226-227).

⁶⁶ Ces standards sont du reste à l'origine des spécifications MIL du ministère américain de la défense.

⁶⁷ Le World-Wide-Web se développe rapidement à partir de 1993. Il existait 800 serveurs WWW en 1994, 650 000 aujourd'hui et le nombre d'utilisateurs a augmenté d'environ 100 % par an depuis six ans (Philippe Dimech et Etienne Turpin, 1998, p.24). Début 2000, le contenu du Web était estimé à 1 milliard de pages, qui correspondrait à 4,2 millions de sites publics selon un pointage signé Inktomi et Nec Research Institute (Les News, 18-1-2000).

Language pour la présentation de l'information, HTTP pour la communication entre ces documents) ont été initialement développés au Centre Européen de Recherche Nucléaire (CERN) par Tim Berners-Lee en 1989 pour faciliter la communication au sein de la communauté de la physique nucléaire par la création d'un système de distribution et d'échange en temps réel d'informations et de données⁶⁸.

Les particularités de ces processus de standardisation (ouverture, évolution maîtrisée collectivement) ont facilité la traduction des différents intérêts de l'ensemble des acteurs très hétérogènes de l'informatique en leur permettant de maintenir leur identité propre. L'aspect fondamentalement décentralisé d'Internet – que l'on retrouve dans la conception du réseau comme des contenus (liens hypermédias en rupture avec les arborescences hiérarchiques traditionnelles de l'informatique) - est en adéquation avec le besoin d'une communication non hiérarchique⁶⁹, permettant une nouvelle maîtrise de l'espace et du temps (non-nécessité de synchroniser les fonctions d'émission et de réception d'informations). En même temps, les évolutions techniques qui accompagnent le développement d'Internet (mise en place d'architectures de type clients-serveurs qui permettent le partage des tâches avec optimisation des ressources entre mainframes, mini-ordinateurs, stations de travail et ordinateurs individuels) vont accélérer la fusion des trois réseaux existants en une structure unique (Gérard Dréan, 1996 A, p. 57).

Le succès de ce nouveau réseau réside également dans sa capacité à fusionner les différents domaines de l'informatique - l'informatique de production (*mainframes*), l'informatique de conception (station de travail) et l'informatique d'aide à la décision (ordinateur personnel) -, et à intégrer les développements du "paradigme numérique" basé sur la numérisation du traitement et de la communication d'informations de toutes natures (écrits, sons, images...) (Godefroy Dang Nguyen et Denis Phan, 1997, p.184). A ce stade, l'informatisation ne permet plus seulement d'automatiser des tâches, mais d'automatiser des

⁶⁸ Le langage HTML a été rendu accessible gratuitement au public en 1992. Le *World Wide Web*; a connu une progression modérée jusqu'à ce que le *National Center for Supercomputing Applications* diffuse gratuitement un logiciel de navigation (Mosaic) permettant aux ordinateurs sous Windows d'interroger les sites connectés sur le Web.

⁶⁹ La conception même du réseau fait qu'il n'existe pas de distinction technique entre producteurs d'informations d'un côté, consommateurs de l'autre, toute personne ou institution connectée au réseau pouvant produire librement des informations reliées facilement (liens hypermédias) aux informations déjà existantes.

relations ou d'établir de nouvelles relations entre les agents économiques⁷⁰. Elle porte directement désormais sur les mécanismes de coordination, internes aux organisations et entre les organisations, ce qui se traduit par une recentralisation partielle de la prise en charge organisationnelle de la fonction informatique au sein de directions des systèmes d'informations (Alain Rallet, 1997, p. 97-101).

Une illustration de ces tendances est la rapidité avec laquelle le protocole TCP/IP s'impose dans les réseaux d'entreprise donnant naissance à des réseaux Intranet⁷¹ qui constituent une réponse à l'hétérogénéité des systèmes distribués, et ouvrent de nouvelles perspectives au "client-serveur". Un réseau Intranet est un réseau informatique privé et protégé, fondé sur les technologies développées pour Internet⁷². Il est composé de messageries, de forums de discussion, de serveurs de fichiers et d'applications, internes à l'entreprise, fournis par un serveur Web, et d'un accès à Internet isolé par un dispositif de protection (*firewall*). Les utilisateurs accèdent à l'ensemble de ces ressources par le biais d'un navigateur (interface qualifiée de client universel) utilisant HTML.

Tableau IV

***Pénétration du protocole TCP/IP dans les réseaux d'entreprises,
en % des matériels connectés***

	<i>PC</i>	<i>Serveurs</i>	<i>Moyens systèmes</i>	<i>Grands systèmes</i>
<i>1996</i>	<i>21</i>	<i>40</i>	<i>44</i>	<i>31</i>
<i>1998</i>	<i>50</i>	<i>50</i>	<i>58</i>	<i>41</i>
<i>1999</i>	<i>67</i>	<i>67</i>	<i>70</i>	<i>44</i>

Source : IDC in 01 Informatique, 14 mars 1997

⁷⁰ Le pourcentage des entreprises industrielles qui transfèrent des données par l'intermédiaire d'une interface informatique est passé de 36,8 % en 1994 à 59,5 % en 1997 (Secrétariat d'Etat à l'Industrie, 1999, p. 69).

⁷¹ Selon Keenvision, le nombre d'entreprises ayant mis en place un Intranet a progressé d'un tiers ces douze derniers mois et atteint plus de 55 % des entreprises de plus de 500 salariés (15 septembre 2000).

⁷² Un réseau Intranet peut s'étendre au-delà des frontières d'une organisation. Par exemple, en France, le projet d'Intranet le plus important est le réseau informatique sanitaire Sésame-Vitale qui devrait connecter l'ensemble des professionnels de la santé, 310 000 médecins libéraux, tous les hôpitaux et toutes les administrations sanitaires et sociales (Eurostaf, 1997 A, p. 24).

La rapidité de cette évolution a pris à contre-pied Novell, qui avec son logiciel Netware basé sur un protocole de communication spécifique (IP Exchange) disposait de la plus grande base installée de serveurs pour réseaux locaux⁷³.

La force des protocoles Internet provient de la nécessité d'un alignement limité des différents acteurs, ce qui lui a permis de regrouper rapidement des acteurs de plus en plus nombreux et surtout de plus en plus divers. Le fonctionnement d'Internet repose sur la capacité des "quelques entités coordinatrices reconnues par la communauté" (Jean-Luc Archimbaud, 1995, p.39), à contrôler et impulser les évolutions techniques, à adopter les standards et à gérer les problèmes d'adresses des machines connectées au réseau. Les principales institutions, apparues au fur et à mesure des besoins sont : l'*Internet Society* (ISOC) qui est, en théorie l'"instance suprême" de l'Internet ; l'*Internet Engineering Task Force* (IETF) qui détermine les évolutions techniques, sous la responsabilité de l'*Internet Architecture Board* (IAB) sorte de "comité des sages" de l'Internet où siègent des scientifiques reconnus pour leurs compétences dans le domaine des réseaux informatiques ; le *World Wide Web Consortium* (W3C) de création plus récente (octobre 1994) qui assure la gestion du World Wide Web ; l'*Internet Assigned Numbers Authority* (IANA) - seule institution centralisée indispensable au fonctionnement de l'Internet - qui gère l'unicité des adresses Internet grâce au "système des noms de domaine" (DNS) : à l'adresse numérique d'une machine raccordée à l'Internet (adresse IP, par exemple 198.105.232.6) est associé un "nom" censé mieux identifier l'entité que l'on cherche à contacter et être plus facile à mémoriser (www.microsoft.com pour l'exemple cité).

Les processus de décisions de ces institutions sont assez particuliers : par exemple, pour élaborer ou faire évoluer un standard, l'IETF publie sur le réseau des *Request For Comments* (RFC) librement accessibles et qui suscitent des échanges en ligne ouverts à tous les participants désireux d'intervenir. La discussion s'arrête quand le groupe ainsi constitué atteint un "consensus approché" (*rough consensus*) sur la définition des spécifications ; à partir de ces spécifications, il est procédé à des essais de prototypes, avec plusieurs prototypes de plusieurs fournisseurs qui doivent être capables de coopérer, et c'est seulement quand il

⁷³ Novell en fort mauvaise posture, ce qui se traduit notamment par une importante vague de licenciements, a du se rallier au protocole IP et tente de se redresser avec la création d'un "annuaire universel" (*Novell Directory System*) qui contient les profils des utilisateurs, les adresses électroniques, les chemins d'accès aux différentes informations. Elle tente d'imposer ce produit comme un standard "ouvert" aux autres entreprises contre un projet identique *Active Directory* de Microsoft.

existe des produits définitifs, avec un choix entre plusieurs fournisseurs, que le standard est définitivement adopté.

Tableau V : Les réseaux technico-économiques dans l'informatique

Réseaux	Informatique traditionnelle	Mini-informatique puis stations de travail	Microinformatique	Fusion tendancielle
Début	Années 60	Années 70	Années 80	Années 90
Intermédiaire déterminant (Standard stratégique)	Architecture matérielle	Système d'exploitation (versions concurrentes d'UNIX)	Système d'exploitation (MSDOS puis Windows)	Protocoles de communication (TCP IP)
Acteur dominant	IBM	Pas d'acteur dominant	Microsoft (et Intel)	Organismes alternatifs
Innovation décisive	Ordinateur	Multiples innovations	Microprocesseur	Client-serveur
Utilisateurs principaux	Administrations et Grandes Entreprises (GE)	Les différents services des administrations et des GE. Les Moyennes Entreprises	Les précédents plus les PME et une partie des ménages	Tous potentiellement
Type d'utilisation	Calcul scientifique Gestion	Conception Création	Outil de productivité personnelle Bureautique Aide à la décision	Très diversifié Outil de communication et de recherche d'informations Automatisation des relations
Valeurs culturelles	Centralisé et élitiste	Expert Décentralisation relative	Démocratique et convivialité	Communication non hiérarchique Réalité virtuelle
Interaction homme / machine	L'homme doit s'adapter à l'informatique	Introduction de la "convivialité"	L'informatique doit s'adapter à l'homme	Multimédia

Le tableau V représente de façon synthétique les quatre réseaux technico-économiques. On remarque que deux de ces réseaux (informatique traditionnelle et microinformatique) sont structurés de façon assez semblable autour d'un acteur dominant (respectivement IBM et Microsoft⁷⁴). Par opposition, les deux autres réseaux ont en commun de ne pas être dominés par un acteur privé. Ils diffèrent fondamentalement selon le degré d'ouverture du standard stratégique : dans le cas du réseau constitué autour d'Unix, l'existence de différentes versions

⁷⁴ Les similitudes entre IBM et Microsoft à trente ans d'écart sont caractéristiques de cette situation d'acteur dominant d'un réseau technico-économique : mêmes croissances fulgurantes et apparemment inexorables, mêmes craintes sur l'étendue de leurs pouvoirs, mêmes critiques virulentes sur les méthodes utilisées pour s'imposer, mêmes démêlés avec les institutions judiciaires...

concurrentes a eu pour conséquence que ce réseau s'est fragmenté en plusieurs sous-réseaux imparfaitement reliés, ce qui a certainement nui au développement du monde Unix, en dépit de ses potentialités techniques. Par contre, dans la situation actuelle, l'ouverture des standards développés par des organismes alternatifs a permis l'extension accélérée de ce réseau à l'ensemble de l'informatique.

Schématiquement les standards stratégiques ont concerné successivement les matériels, les systèmes d'exploitation, les formats de présentation et les protocoles de communication. Chaque déplacement du caractère décisif de la standardisation d'un intermédiaire à un autre s'accompagne de la possibilité de bénéficier de la diversité sur l'intermédiaire précédent, tout en bénéficiant des externalités de réseaux sur le nouvel intermédiaire : par exemple, la standardisation au niveau du système d'exploitation autorise l'utilisation de tous les matériels sur lesquels est implémenté le système standard ; la standardisation au niveau des protocoles de communication permet d'interconnecter tous les systèmes informatiques - composés de matériels et de systèmes d'exploitation très divers – qui supportent ces protocoles de communication.

Le rapprochement entre les trois mondes de l'informatique dépasse leur simple interrelation technique. En s'effectuant sur des standards "ouverts", il a pour conséquence que pour chaque élément fonctionnel, plusieurs produits se retrouvent en concurrence, tout en étant compatibles avec le reste du système (Gérard Dréan, 1996 A, p. 311). De ce fait, c'est l'ensemble de l'industrie informatique (et plus seulement la micro-informatique) qui éclate en secteurs spécialisés, à l'intérieur desquels les représentants des trois mondes se retrouvent en concurrence : "dans chacun de ces secteurs spécialisés, la concurrence fait disparaître les frontières entre les trois mondes du PC, des ordinateurs traditionnels et des systèmes ouverts, et a pour effet d'unifier globalement l'informatique" (idem, p. 286).

Cette évolution du niveau stratégique de la standardisation s'est accompagnée d'une diversité croissante des utilisateurs de l'informatique, de leurs modes de communication avec les ordinateurs et des utilisations de l'informatique. La prolongation de cette tendance - communications sous des formes de plus en plus variées entre utilisateurs reliés, connexions avec des objets non informatiques - va nécessiter une nouvelle évolution des standards selon un processus qui reste à déterminer mais pour lequel l'histoire passée peut être riche d'enseignements.

B - QUELLES PERSPECTIVES ?

S'appuyant sur la croissance accélérée d'Internet et des réseaux Intranet, des acteurs informatiques importants tentent d'opérer une reconfiguration de l'ensemble de l'informatique autour de l'idée de Network Centric Computing, formulée par Larry Ellison (PDG d'Oracle)⁷⁵. Ce modèle est basé sur trois maillons : *Fat Server*, *Big Pipes*, *Thin Client*. Il est principalement soutenu par Sun, leader des stations de travail et qui produit des serveurs de plus en plus puissants sous Unix, utilisant dès l'origine TCP/IP, par Oracle, leader des SGBDR (Système de Gestion de Bases de Données Relationnelles) et par Netscape (navigateur et logiciel de gestion de serveurs). Il est fondé sur l'existence d'un langage universel, Java développé par Sun⁷⁶, qui permet d'écrire des applications s'exécutant sur toutes les machines disposant d'un programme d'interprétation intégré aux navigateurs et aux serveurs (machine virtuelle Java). De ce fait, aux PC pourraient être avantageusement substitués des NC (*Network Computer*), terminaux administrés à distance et tirant ses ressources du réseau (Intranet ou Internet). En effet, le PC réclame toujours plus de puissance et a un coût élevé : au-delà de l'investissement d'origine en matériel et logiciels estimé à 7 800 dollars, le coût de propriété d'un PC (TCO *Total Cost of Ownership*) est évalué par le Gartner Group en 1966 à 39 000 dollars sur cinq ans (Eurostaf, 1997 A, p. 55) et selon la même source, le Network Computer permettrait de baisser ce coût de 34 % (01 Réseaux, Juin 1998).

Pourtant ce nouveau modèle informatique a du mal à s'imposer. Les problèmes techniques fréquemment avancés (débit limité des infrastructures de communication, performances insuffisantes des machines virtuelles Java pour exécuter des grandes applications) n'expliquent que très partiellement la modestie des ventes de NC (contrairement

⁷⁵ En réalité, on peut considérer que l'origine de cette idée remonte aux débuts de la micro-informatique en 1981, avec le projet (jamais concrétisé) de Digital Research de fabriquer avec Intel des « *diskless machines aimed at networked applications* » (Datamation, janvier 1982, p. 200).

⁷⁶ Le succès de Java a été tel qu'il a conduit IBM à abandonner *Opendoc*, une technologie objet développée avec Apple et Novell, qui ne servira plus que de "pièces détachées" pour les composants de Java (*javabeans*). Java, qui au départ était un simple langage de programmation a été transformé par Sun en une architecture de développement complète d'applications (JDK) (Eurostaf, 1997 A, p.45-47).

à l'ensemble des prévisions⁷⁷), et leur résolution prévisible restera insuffisante⁷⁸. Une analyse en terme de réseaux technico-économiques peut permettre d'expliquer cet échec prévisible.

Premièrement cette conception s'oppose directement et frontalement aux intérêts de deux des acteurs les plus puissants de l'informatique actuelle : Intel pour les microprocesseurs, auquel Sun veut opposer un microprocesseur spécifique à Java (PicoJava), et Microsoft dont le système d'exploitation Windows pourrait être concurrencé par JavaOS. Leur réaction n'a pas tardé : pour imposer son navigateur Internet Explorer contre la position hégémonique détenue par Netscape, Microsoft l'a distribué gratuitement avant de l'intégrer dans Windows 98 (ce qui est la cause de son procès actuel) ; surtout Microsoft a ajouté au langage HTML et à Java ses propres caractéristiques⁷⁹ ce qui a pour conséquence que certaines pages Web s'affichent différemment voire refusent de s'afficher suivant le navigateur utilisé, ou que les serveurs sont obligés de développer deux fois leurs sites. Pour résoudre les problèmes d'interfaces entre des objets (logiciels) répartis sur différentes machines, les principaux acteurs informatiques ont adopté le standard CORBA auquel Microsoft oppose DCOM⁸⁰. Cette situation de "déstandardisation" a pour conséquence de nourrir un certain attentisme notamment dans la réécriture des applications en Java, nécessaire pour être utilisées sur des NC. Par ailleurs, Microsoft et Intel en alliance avec Compaq, Hewlett-Packard et d'autres constructeurs moins importants s'emparent (en la dénaturant) de l'idée de *Thin Client* en créant des terminaux Windows, qui gèrent uniquement l'interface graphique, le traitement applicatif étant réalisé sur un serveur sous Windows NT, et des NetPC, qui sont des PC où certains éléments permettant d'introduire des données ont été supprimés ou verrouillés. Pour

⁷⁷ Les prévisions qui allaient de plusieurs centaines de milliers à plusieurs millions de NC vendus en 1998 (Eurostaf, 1997 A, p. 18-19) ont été démenties par des ventes demeurées marginales.

⁷⁸ Plus récemment Larry Ellison (PDG d'Oracle) a quelque peu modifié son projet vers le développement d'appareils alternatifs au PC permettant de se connecter à Internet et dont il prévoit que le nombre dépassera celui des PC connectés à Internet dès l'an 2000 ! Oracle a créé une filiale « Network Computer » pour produire de tels appareils, dont le point fort devrait être une facilité d'utilisation, identique à celle d'une console de jeu vidéo, et qui seraient vendus au prix de 199 dollars, et a entamé des discussions avec une compagnie de téléphone européenne afin de mettre au point des téléphones offrant des fonctionnalités proches de celles des ordinateurs de réseaux (18h.com, Le quotidien de l'Expansion, 28/10/99).

⁷⁹ Le navigateur Internet Explorer 5.0 de Microsoft n'est pas conforme aux standards définis par le W3C alors que Microsoft a participé à l'élaboration de ces standards (01 Informatique, 26-3-99).

⁸⁰ Microsoft participe pourtant au consortium qui développe CORBA, mais d'une façon assez similaire à IBM, qui participait activement aux instances de normalisation tout en tentant par ailleurs d'imposer sur le marché ses solutions comme des standards de fait.

également réduire l'avantage en termes de coût du NC, différentes initiatives ont été prises pour faire baisser les coûts d'administration des PC : Microsoft a développé une solution logicielle (*Zero Administration Kit*) ainsi qu'IBM (*System Case*), Dell a intégré à son système de vente directe de micro-ordinateurs constitués de composants choisis par le client (*Build To Order*), la possibilité d'une configuration sur mesure de l'ordinateur à la commande. Ces différentes actions ont permis d'entraver le développement des NC et de limiter son avantage économique.

Deuxièmement, et cause principale probable du manque d'avenir des NC, la conception "*Network Centric Computing*" n'est pas en adéquation avec les aspirations socioculturelles dominantes des utilisateurs de l'informatique. Elle répond certes aux besoins de communication mais, en réintroduisant une vision hiérarchique (qui n'est pas sans rappeler l'architecture verticale de l'informatique traditionnelle), elle s'oppose à la volonté d'autonomie des utilisateurs.

Si l'on peut en conclure que cette conception ne deviendra pas le modèle informatique de demain, demeure la question de l'évolution du processus de standardisation, initié par des institutions spécifiques (ni privées, ni étatiques) et qui devient de plus en plus complexe à mettre en œuvre, au fur et à mesure que le groupe des utilisateurs se multiplie et s'hétérogénéise, et que l'informatique est utilisée pour des activités de plus en plus diverses (commerce électronique⁸¹, travail coopératif réparti assisté par ordinateur...).

La question se pose du "degré de stabilité et d'autonomie à terme d'un organisme alternatif (...) et de la probabilité non négligeable qu'un tel organisme sous la pression économique ou politique soit capturé par le système public ou par le système privé, par quelques grands acteurs économiques recherchant à promouvoir leurs propres technologies" (Pierre-José Billotte, 1997, p. 203). La maîtrise de l'évolution des standards devient une question stratégique décisive, et les alliances⁸² et les rapprochements entre fournisseurs de "contenus", opérateurs de télécommunications, producteurs de matériels et de logiciels, et nouvelles sociétés de services liées à Internet (fournisseurs d'accès, de sites "portail", de

⁸¹ Dans ce domaine, 22 grandes entreprises multinationales ont constitué le *Global Business Dialog* (GBD) dont l'objectif est de promouvoir des règles pour le commerce électronique.

⁸² Même si l'on peut douter de la pérennité de certains accords dont l'objectif est plus d'influencer les anticipations par leur effet d'annonce que de se réaliser concrètement.

moteurs de recherches...) se multiplient. Par exemple, pour la question des paiements via Internet, en 1995 Microsoft et Visa proposaient le standard *Secure Translation Technology* en le présentant comme "ouvert" (disponible pour toute entreprise souhaitant l'employer), mais en réalité le logiciel nécessaire pour rendre le standard opérationnel ne pouvait être obtenu que par un accord de licence auprès de Visa et de Microsoft ; ce standard s'opposait au standard *Secure Courier* soutenu par MasterCard, Intuit, IBM et Netscape ; cette compétition s'est toutefois achevée par l'adoption d'un standard commun *Secure Electronic Transactions* en février 1996 (Carl Shapiro, Hal R. Varian, 1999, p. 222-223).

Jusqu'à maintenant les institutions d'Internet ont résisté aux différentes tentatives de "privatisation" des standards : passage d'IP version 4 à IP version 6 pour faire face à la pénurie potentielle d'adresse⁸³, développement d'IP Security pour résoudre les problèmes de sécurité, naissance de la version 4 d'HTML intégrant les nouvelles fonctionnalités nécessaires au multimédia tout en tentant d'empêcher la fragmentation du marché et de garantir l'interopérabilité. Le plus délicat fut l'évolution de la gestion des noms de domaine : d'une question essentiellement technique résolue par l'IANA (institution qui n'avait pas d'existence juridique et qui était principalement animée par un des fondateurs d'Internet, Jonathan Postel, décédé récemment), elle est devenue un enjeu important pour les entreprises introduisant des aspects complexes de propriété intellectuelle et industrielle. Après de multiples péripéties, les conflits semblent s'estomper avec la création en 1998 d'une nouvelle institution l'*Internet Corporation for Assigned Names and Numbers* (ICANN). Une autre difficulté vient de l'extension de l'utilisation d'Internet pour la diffusion de l'audio et de la vidéo où la plupart des standards appartient à des entreprises privées (par exemple, Thomson Multimédia détient les droits du format de compression MP3). "Le danger du Net, aujourd'hui, vient de la multiplication des standards, parce que ce sont des entreprises et non les milieux académiques qui développent la plupart des innovations (transmission de la voix, de la vidéo, etc.). Il faut des organisations très puissantes pour maintenir la disponibilité la plus large possible du système dans de telles conditions" estime Robert Cailliau, co-inventeur du Web avec Tim Berners-Lee (Chronique de Cybérie ; 10 août 1999).

⁸³ Ce qui a permis de démentir la prévision de Sandra Braman selon laquelle, "l'accroissement du nombre de personnes souhaitant accéder à Internet est en soi devenu une marchandise, étant donné que les adresses IP nécessaires pour identifier individuellement chacun des utilisateurs deviennent des biens rares et commercialisables dont le prix va probablement augmenter" (1997, p.105).

Toutefois, face aux évolutions techniques, la pérennité de cette situation n'est nullement garantie. Se profile une reconfiguration élargie du réseau qui pourrait voir l'extension des chaînes de traduction à l'ensemble des objets qui contiennent des microprocesseurs (l'ensemble des appareils électroménagers, les téléphones, les décodeurs, l'automobile, les consoles de jeux, les multiples cartes à puces, les agendas électroniques...) ⁸⁴. Si la plupart de ces objets ont été à l'origine conçus selon une logique câblée (où les instructions sont gravées dans le silicium), c'est une logique programmée avec l'intégration de véritables logiciels qui se développe actuellement ⁸⁵, suivant la même tendance que celle que l'on a observée dans l'informatique (Jean-Benoît Zimmermann, 1998 A).

Elle ouvre la possibilité pour l'ensemble de ces objets de s'intégrer dans les réseaux informatiques en permettant la communication entre ces objets divers et les ordinateurs, générant un vaste champ de nouvelles applications (*appliances*) encore très floues (la domotique notamment) et développant de nouvelles externalités de réseaux. La création de ce nouveau réseau nécessite un "alignement" relatif de ces objets, de leurs producteurs et de leurs utilisateurs. Les questions de standardisation pour résoudre les problèmes de communication et pour permettre à des applications de s'exécuter partiellement sur des objets aussi divers (informatique diffuse) se posent dans des termes différents et concernent de nouveaux acteurs ⁸⁶. A l'heure actuelle, Microsoft tente d'imposer comme système d'exploitation Windows CE, version allégée de Windows 95, qui est confronté notamment à eCos développé par la société Cygnus et compatible avec la norme japonaise Itron. Sun, qui a développé avec succès le langage Java, même s'il n'a pas réussi à en faire la base de l'ensemble de l'architecture informatique, essaye de rééditer cette opération pour l'ensemble des objets avec la création de Jini. Si les solutions proposées par Microsoft sont complètement "privées", les autres solutions, notamment celles de Sun, présentent un certain degré d'ouverture : publication et libre disponibilité des spécifications et du code-source des

⁸⁴ Sur les 6 milliards de microprocesseurs dans le monde, moins de 600 millions se trouvent dans des micro-ordinateurs.

⁸⁵ Outre sa plus grande souplesse (possibilité de reprogrammer) la logique programmée peut permettre grâce à la standardisation des composants de générer de formidables économies d'échelle.

⁸⁶ Par exemple, Sony a affiché l'objectif de devenir un acteur majeur dans la réalisation de systèmes d'exploitation et d'applications pour appareils ménagers, ce qui s'est traduit par l'annonce de nombreuses embauches d'ingénieurs spécialisés dans le logiciel et la reconversion d'une partie de ses salariés spécialisés dans le matériel.

logiciels. Toutefois selon un modèle qui semble se développer en informatique, Sun garde un contrôle important par le dépôt de brevets⁸⁷ et par son pouvoir d'accorder une certification de compatibilité.

Les enjeux sont donc considérables. L'analyse des créations précédentes des différents réseaux technico-économiques laisse supposer qu'une convergence se produira sous des formes qui restent à déterminer, l'histoire de l'informatique ayant montré l'importance que pouvaient prendre des événements et des décisions apparemment mineurs dans un contexte particulier propice à la naissance d'un nouveau réseau. Le deuxième enseignement que l'on peut dégager de l'étude de l'émergence des standards déterminants en informatique est que ce processus peut s'opérer selon deux modalités très différentes. La première possibilité est que la convergence s'opère sur des standards privés (sur le modèle de l'informatique traditionnelle et de la microinformatique) imposés par un acteur déterminant de l'informatique ou plus vraisemblablement par une coalition d'acteurs privés appartenant aux différents domaines concernés. Outre le fait qu'il est probable que ce processus prendra un certain temps et passera par une phase de "balkanisation" du réseau, résultant de la coexistence de différents standards concurrents et incompatibles, avant qu'une dynamique d'autorenforcement se crée autour d'une solution qui ne sera pas nécessairement optimale, ce scénario a l'inconvénient de conférer à la coalition d'acteurs qui contrôle les standards des pouvoirs exorbitants⁸⁸ sur l'évolution de l'informatique, voire sur une part de plus en plus importante de l'ensemble de l'activité économique et sociale. La deuxième possibilité, qui peut sembler plus efficace quant à l'émergence d'un nouveau réseau et plus satisfaisante sur un plan démocratique, est que ce réseau se structure autour de standards ouverts portés par des organismes alternatifs, suivant le modèle qui a été à la base du succès d'Internet.

⁸⁷ En Europe, à la différence des Etats-Unis, il n'est pas possible de breveter des logiciels. Ceux-ci relèvent du droit d'auteur avec quelques aménagements (les droits d'auteur sont possédés par l'employeur et non par l'auteur comme dans les autres œuvres). Toutefois la jurisprudence a ouvert, dans certains cas, la possibilité de breveter des logiciels, et des propositions existent au niveau européen pour adopter le modèle américain de brevetabilité complète des logiciels (cf. chapitre III).

⁸⁸ Ces pouvoirs peuvent être de nature économique (possibilité de "péages" dans une position de "garde barrière") mais aussi sociale (influence sur les contenus diffusés, protection de la vie privée...). L'émotion suscitée par la découverte de numéros d'identification, dans le dernier microprocesseur (Pentium III) d'Intel et dans les logiciels de Microsoft, permettant de connaître l'activité des utilisateurs, en est une illustration.

Nous avons analysé l'histoire de l'informatique comme la constitution successive de réseaux technico-économiques structurés par des standards. Nous avons vu que les processus de standardisation pouvaient être l'œuvre d'acteurs divers (entreprises, Etat mais également organismes alternatifs), ne pas être nécessairement optimaux et, malgré tout perdurer, permettant la consolidation et l'extension d'un réseau technico-économique, qui en retour renforce ces standards. Toutefois, dans un contexte marqué par la très forte croissance de l'informatique, l'existence d'innovations techniques et de nouveaux besoins, correspondant notamment à des changements culturels, a permis l'émergence de nouveaux réseaux technico-économiques, qui coexistent avec les anciens, la situation actuelle étant marquée par une fusion tendancielle entre ces différents réseaux.

On ne peut comprendre les évolutions de l'économie du logiciel en faisant abstraction de cette histoire. C'est l'histoire de ces réseaux, dans lesquels s'insèrent les acteurs de l'économie du logiciel, qui permet d'expliquer l'apparition de nouveaux producteurs de logiciels, le déclin d'entreprises qui apparaissaient toutes puissantes, les transformations de la nature économique des produits logiciels... En même temps, l'économie du logiciel joue un rôle de plus en plus important dans l'histoire de l'informatique, ce qu'illustre le fait que les standards déterminants se sont déplacés du matériel vers les logiciels.

Chapitre III

LE LOGICIEL :

UN PRODUIT ET UNE VALEUR "INSAISSABLES" ?

Le logiciel apparaît par bien des aspects comme un produit particulièrement difficile à appréhender. Il l'est tout d'abord en raison de son extraordinaire diversité fonctionnelle. Les logiciels pénètrent les domaines d'activité les plus variés et sont confrontés à des problèmes de plus en plus complexes, auxquels il semble que l'informatisation, grâce à l'augmentation de la puissance des matériels et à la baisse de leur coût, puisse apporter une solution. En même temps, dans certains cas l'informatisation d'une situation peut contribuer à la complexifier davantage, ce qui nous amènera à nous interroger sur les significations multiples de la notion de complexité. Dans l'économie des logiciels se produisent de multiples innovations qui concernent l'ensemble des dimensions de la production des logiciels : apparition de nouveaux produits, de nouvelles méthodes de production, utilisation de nouveaux outils. Si parfois les changements peuvent être extrêmement rapides (par exemple un logiciel de navigation sur Internet est devenu un outil banal et largement utilisé alors qu'il n'existait pas il y a quelques années), il faut également prendre en compte le temps et les efforts d'assimilation et d'apprentissage, qui expliquent que la diffusion de certaines innovations puisse être beaucoup plus lente. Ces évolutions qui se produisent sur une période très courte (la production de logiciel comme une activité distincte n'a pas plus de trente années d'existence) expliquent l'instabilité technologique, qui semble caractéristique de l'économie du logiciel et qui rend plus difficile son analyse.

L'analyse n'est pas plus facile sur un plan économique. Qu'est-ce économiquement qu'un logiciel ? S'agit-il d'un bien ou d'un service ? Est-ce un produit sur lequel peuvent s'exercer des droits de propriété ? A quelles conditions techniques et/ou juridiques ? Qu'est-ce qui détermine son prix ? Les théories de la valeur des économistes peuvent-elles lui être appliquées ?

Pour répondre à ces questions, nous verrons comment les spécificités technico-économiques des logiciels rendent difficile l'utilisation des catégories habituelles des

économistes (section II). Auparavant, nous étudierons les grandes lignes de l'évolution technologique extrêmement rapide, caractéristique de l'économie du logiciel (section I). La séparation de "l'insaisissabilité" des logiciels en une dimension plus technique et une dimension plus théorique a été opérée pour faciliter l'exposition des raisonnements, ces deux dimensions étant intimement liées dans la réalité.

Section I - L'"insaisissabilité" technique : l'évolution

extrêmement rapide des produits, de leurs domaines d'utilisation et des techniques de production

L'économie du logiciel connaît des changements permanents qui correspondent à l'ensemble des modèles d'innovations mis en évidence pour l'ensemble de l'économie (C). Ces innovations sont une réponse à la complexité croissante des problèmes que doivent résoudre les logiciels (B), sur une période relativement brève, l'économie du logiciel ne s'étant constituée comme activité économiquement séparée que depuis une trentaine d'années (A).

A - LA JEUNESSE DE LA PRODUCTION DE LOGICIELS COMME ACTIVITE SEPARÉE

Si la production d'ordinateurs a environ un demi-siècle d'existence (elle commence après la seconde guerre mondiale), la production de logiciels comme une activité séparée ne date que du début des années soixante-dix¹. Certes, l'utilisation des ordinateurs a toujours nécessité l'existence d'une activité de programmation², mais celle-ci était exclusivement effectuée par les utilisateurs, les premiers ordinateurs étant fournis "nus". Au début des années soixante, les matériels étaient livrés avec des logiciels (principalement des systèmes d'exploitation et des langages de programmation) mais seul le matériel faisait l'objet d'une facturation.

¹ "L'informatique est une industrie née il y a cinquante ans, et pourtant la question du logiciel n'a pas été posée pendant les vingt-cinq premières années, bien que celles-ci aient été caractérisées par une immense créativité logicielle" (Christian Genthon, Denis Phan, 1999, p. 172).

² Le premier livre écrit sur la programmation, à partir des travaux de l'équipe de l'EDSAC, date de 1951 (Jean-Yvon Birrien, 1990, p. 53).

L'émancipation économique du logiciel par rapport au matériel a résulté de plusieurs phénomènes :

- Au milieu des années soixante, IBM décide de produire une série d'ordinateurs (les 360) compatibles entre eux et devant couvrir l'ensemble des besoins. La mise en application de cette décision qui aura un impact décisif sur le développement de l'informatique traditionnelle et son contrôle par IBM (cf. chapitre II), connaît de sérieuses difficultés dans sa partie logicielle : retards dans les développements, nécessité de fournir des services importants d'assistance et de formation pour résoudre les problèmes rencontrés par les utilisateurs (Gérard Dréan, 1996 A, p. 31). De simple produit complémentaire à la vente de matériel, le logiciel devient "un poste de dépenses majeur et un facteur concurrentiel souvent déterminant pour le constructeur" (idem, p. 19). D'autre part, la non-facturation des services de plus en plus substantiels pose des problèmes nouveaux, comme la question de savoir si IBM devait continuer à assurer la maintenance des matériels IBM fournis par des revendeurs d'occasion qui prenaient de plus en plus d'importance.

- Dans le même temps, en réponse à la demande de logiciels applicatifs et de services de plus en plus diversifiés, se développent des Sociétés de Service et de Conseil en Informatique (S.S.C.I.), qui, si elles facturent leurs prestations, présentent l'avantage d'être (relativement) indépendantes des constructeurs et de disposer de compétences spécifiques, dans une période de pénurie de personnels qualifiés (Eric Labat, 1984, p. 195).

- Enfin, IBM se trouve sous la pression des procédures judiciaires au titre des lois anti-trust.

C'est dans ce contexte qu'IBM adopte en 1969 une politique d'*unbundling* ou facturation séparée des matériels, des logiciels, de l'assistance technique et de la formation, en anticipant un mouvement qui paraissait inéluctable (Jean-Michel Treille, 1973, p. 97). Dans un premier temps seule une partie des services et du logiciel devient facturable, l'autre partie restant incluse dans le prix du matériel. Pour les logiciels seuls les logiciels applicatifs non essentiels à l'exploitation (*Program Products*) sont facturés séparément, les logiciels "nécessaires à l'exploitation" (*Systems Control Programs*) restant compris dans le prix du matériel. Cette disposition ne sera que transitoire et à partir des années quatre-vingt, tous les logiciels fournis par IBM sont facturés. La décision d'*unbundling* marque la naissance d'une économie du logiciel indépendante, en permettant notamment l'essor des SSCI. Ce mouvement s'amplifiera avec le développement des progiciels à partir des années quatre-vingt. Il est intéressant de

noter que si certaines évolutions récentes semblent prendre la direction inverse en proposant des solutions informatiques globales, intégrant des composants matériels et logiciels, elles ne remettent pas en cause l'autonomie de l'économie du logiciel, les fournisseurs de ce type de solutions achetant la plus grande part des logiciels qu'ils intègrent.

C'est donc sur une période très courte (trente ans) qu'il faut analyser des évolutions correspondant schématiquement au passage de l'artisanat à la grande industrie automatisée, un passage qui a pris plusieurs siècles dans d'autres secteurs. En se référant aux trois grandes étapes de l'évolution professionnelle dans l'industrie mises en évidence par Alain Touraine (1962), on peut dire que la production des logiciels est passée rapidement de l'étape A (prédominance de l'action autonome de l'ouvrier qualifié, en l'occurrence l'informaticien) à l'étape B (prédominance de l'organisation centralisée du travail intégrant des analystes, des programmeurs...), et qu'elle entre déjà dans l'étape C de l'automatisation de la fabrication avec le développement d'outils de génie logiciel. En même temps la nature des problèmes que doivent résoudre les logiciels a considérablement évolué.

B - LA COMPLEXITE CROISSANTE DES PROBLEMES A RESOUDRE

Un large accord existe sur la complexité croissante des problèmes que doivent résoudre les logiciels, et l'accroissement de la complexité des logiciels qui en résulte³. Le logiciel est plus complexe et plus difficile à maîtriser que le matériel⁴. La critique, fréquente, de cette situation, omet le fait que c'est dans la nature même du logiciel de réaliser ce qui est trop complexe pour le matériel (Gérard Dréan, 1996 A, p. 198). Une idée nouvelle ou l'exploration d'un nouveau domaine se traduit d'abord par le développement de programmes. Ce n'est que lorsque la complexité de cette nouvelle application aura été maîtrisée qu'il deviendra éventuellement possible et souhaitable d'inscrire certaines des fonctions nécessaires sur des composants électroniques spécialisés.

³ Par exemple, Patrick Jaulent estime que "l'un des traits les plus marquants de ces dernières années en matière de logiciel est l'accroissement de la complexité des applications réalisées" (1992, p. 18). Frederick P. Brooks considère que "les entités logicielles sont sans doute les constructions les plus complexes réalisées par l'homme" (1996, p. 158).

⁴ "La description externe d'un système logiciel est 10 à 20 fois plus longue que celle de l'ordinateur proprement dit" (Frederick P. Brooks, 1996, p. 35).

Au-delà de ce constat, il est nécessaire de définir précisément ce qu'est la complexité (1), avant de voir que la complexité croissante des logiciels est tout à la fois le produit d'un environnement informatique de plus en plus complexe (2) et de la volonté d'automatiser des fonctions de plus en plus complexes (3). Enfin si les logiciels doivent intégrer cette complexité croissante, il faut également tenir compte qu'en retour eux-mêmes peuvent être créateurs de complexité (4).

1 - Les significations multiples de la notion de complexité

La complexité n'est pas simple à définir... La complexité peut prendre plusieurs significations différentes mais liées. La complexité peut évoquer les difficultés pour résoudre un problème (considéré comme complexe). Cette complexité que Faïz Gallouj (1994, p. 196) qualifie de *complexité-compréhension*, est relative aux compétences de la personne ou de l'organisation.

La complexité peut renvoyer à l'incertitude, qui résulte de la méconnaissance et/ou du caractère imprévisible de certains phénomènes, et qui empêche d'avoir une bonne vision du comportement d'un système qualifié de complexe. La fréquence et l'importance des changements peuvent accroître cette complexité, désignée comme étant une *complexité-mouvement* (Faïz Gallouj, 1994, p. 196).

En privilégiant une dimension quantitative, la théorie de l'information définit la complexité comme une quantité d'information, liée au nombre et à la variété des éléments d'un système⁵, dont la combinatoire détermine le nombre d'états qu'il peut prendre⁶. En ce sens un système complexe comporte beaucoup de pièces très différentes, ce qui le rend difficile et coûteux à assembler, et il regroupe une quantité d'information élevée puisque chaque pièce est un cas particulier. Cette vision de la complexité renvoie à la notion de *complexité-volume* de Faïz Gallouj.

⁵ Or, normalement, un logiciel bien construit n'a pas deux parties identiques (du moins au-dessus du niveau de l'instruction). Quand il existe deux parties identiques, elles sont fusionnées en une seule sous forme d'un sous-programme qui sera appelé plusieurs fois. De ce point de vue, un logiciel est différent d'un ordinateur, d'un bâtiment ou d'une automobile qui comprennent beaucoup d'éléments en de multiples exemplaires (Frederick P. Brooks, 1996, p. 158).

⁶ Les ordinateurs sont déjà d'un niveau de complexité extrêmement élevé car ils ont un très grand nombre d'états. Les systèmes logiciels ont un nombre d'états supérieur de plusieurs ordres de grandeur à celui des ordinateurs (Frederick P. Brooks, 1996, p. 158).

Pour l'analyse systémique, ce qui caractérise la complexité c'est moins le nombre de composants différents d'un système que l'importance des interactions entre ces éléments. Dans un système complexe "le tout est plus que la somme des parties" et "étant donné les propriétés des parties et les lois de leurs interactions, l'inférence des propriétés du tout n'est pas une question triviale" (Herbert Simon, 1974, p. 106-107). C'est ce qui explique que pour construire l'intelligibilité d'un système complexe, il soit nécessaire de le modéliser (processus de compréhension), alors que l'intelligibilité d'un système compliqué peut être découverte par simplification (processus d'explication) (Jean-Louis Le Moigne, 1990, p. 11). Jean-Louis Le Moigne (1977) définit neuf niveaux de complexité, classés dans l'ordre de complexité croissante des objets :

Niveau 1 : L'objet est passif.

Niveau 2 : L'objet est actif. Il transforme des flux de façon prédéterminée (processeur).

Niveau 3 : L'objet actif comporte un processeur de régulation.

Niveau 4 : L'objet est capable de capter de l'information et de l'utiliser dans son fonctionnement.

Niveau 5 : L'objet devient capable de décisions. (possède un processeur décisionnel autonome).

Niveau 6 : L'objet possède une mémoire qui permet à son processeur actif et à son processeur décisionnel de fonctionner en tenant compte non seulement du comportement actuel mais également des comportements passés.

Niveau 7 : Chacun des processeurs (actif, décisionnel, mémoire) est lui-même un système complexe. Ils sont appelés respectivement Système Opérant, Système de Pilotage, Système d'Information et leur coordination est indispensable.

Niveau 8 : L'objet est capable d'auto-organisation, d'où la nécessité de la présence d'un système d'imagination-conception dans le système de pilotage.

Niveau 9 : L'objet s'autofinalise (présence d'un système de finalisation). L'objet peut définir ses objectifs et les faire évoluer.

Enfin, les informaticiens utilisent la notion de *complexité algorithmique* qui étudie la relation entre la taille d'un problème, et le volume de mémoire et/ou le nombre de pas de calcul qui sera nécessaire pour le résoudre. Cette complexité algorithmique peut être appréhendée de deux façons différentes. La première est une mesure de la performance des systèmes qui prend en compte les contingences des machines réelles. La seconde, de nature plus théorique et d'essence mathématique, est une fonction qui détermine le nombre d'instructions élémentaires abstraites nécessaires pour exécuter l'algorithme en fonction du nombre d'éléments qui caractérisent les données en entrée de l'algorithme. En pratique, on ne s'intéresse qu'au terme d'ordre le plus élevé et on néglige les facteurs constants. Par exemple, un algorithme de tri dont la complexité est en $O(n^2)$ nécessite un ordre de grandeur de n^2 opérations pour trier n éléments. Les meilleurs algorithmes de tri sont de complexité $O(n \log n)$

n) ; ils sont préférables au précédent puisque quand n (nombre d'éléments à trier) augmente, n Log n (nombre d'instructions à exécuter) croît moins rapidement que n^2 . Il existe deux grandes classes d'algorithmes : ceux dont la complexité s'exprime sous la forme d'un polynôme et ceux dont la complexité ne peut être bornée par un polynôme (NP *Non Polynomial*, car contenant des exponentielles). Dans ce dernier cas, l'augmentation du nombre d'éléments en entrée sature très rapidement l'ordinateur le plus puissant et l'algorithme est en pratique inutilisable. Les problèmes de nature NP sont très fréquents⁷ (notamment les problèmes d'optimisation) et peuvent avoir une apparence anodine : l'exemple le plus célèbre est celui du voyageur de commerce qui veut minimiser son trajet pour parcourir n villes. Dans ces situations l'informatisation du problème nécessite de recourir à des heuristiques. A la différence des algorithmes dont on peut démontrer qu'ils donnent la solution optimale, les heuristiques sont des raisonnements formalisés (en général moins coûteux en nombre d'instructions) dont on tient pour plausible qu'ils donneront une solution satisfaisante (*satisfactum* ou *satisficing*, Herbert Simon, 1974, p. 83) du problème.

En conclusion, quel que soit le sens que l'on attribue à la notion de complexité, il apparaît que les logiciels doivent répondre à des problèmes de complexité croissante, ce que permet l'amélioration des performances du matériel.

2 - Amélioration des performances du matériel et environnement informatique de plus en plus complexe

C'est en effet, l'augmentation exponentielle de la puissance du matériel qui permet l'utilisation de logiciels de plus en plus complexes. Sa meilleure illustration est la "loi" de Gordon Moore (cofondateur d'Intel) énoncée à la fin des années soixante, selon laquelle la puissance des microprocesseurs et la capacité des mémoires doubleraient tous les deux ans (révisée ensuite en tous les 18 mois) : trente ans de vérification de cette loi équivalent à un rapport prix/performances multiplié par un million⁸.

⁷ "Par malchance pour les ordinateurs, les algorithmes intéressants pour les humains ont une fâcheuse tendance à avoir une complexité élevée (...). Pour ce type de situation, il peut ne pas exister d'algorithmes, soit parce que l'algorithme est inconnu, soit parce que l'algorithme requiert des ressources de calcul impossibles à réunir" (Jacques Printz, 1998, p. 29-33).

⁸ Michel Catinat souligne qu'il "n'existe pas d'autre exemple où le progrès technologique est comparable en intensité et en durée". Les performances des technologies informatiques, à savoir une vitesse de traitement et une capacité de stockage multipliées par 100 tous les 10 ans depuis les années soixante, donneraient dans le transport par rail les résultats suivants : la durée d'un trajet Paris Marseille serait passée de dix heures dans la

La complémentarité économique entre les producteurs de matériels et de logiciels est forte, ce que traduit l'expression "Wintel" (Windows + Intel) qui désigne l'alliance objective existant dans la micro-informatique entre les intérêts d'Intel pour les microprocesseurs et de Microsoft pour les logiciels. L'augmentation de la puissance des matériels suscite la production de logiciels plus "gourmands" en ressources matérielles. En retour, le développement de ces logiciels incite les utilisateurs à renouveler leurs matériels.

Le rapport dynamique qui lit le matériel et le logiciel est également un rapport technique concernant leur production. Ce sont les progrès du matériel informatique qui ont, entre autres facteurs, permis de développer des logiciels plus complexes. Réciproquement ce sont les progrès dans les logiciels notamment de CAO et de DAO qui ont permis de produire ces composants de plus en plus puissants (Jean-Louis Cacomo, 1996, p. 31).

La complexité croissante des systèmes informatiques résulte aussi de la diversité grandissante des multiples périphériques d'entrée et de sortie d'informations qui ont vu le jour. Ici également, les interdépendances sont fortes. C'est l'amélioration du rapport prix/performance des composants matériels de base et les perfectionnements des logiciels qui ont rendu techniquement possibles et économiquement viables l'utilisation de ces périphériques, qui facilitent les communications avec un système informatique. Mais l'emploi de ces périphériques nécessite des ressources matérielles supplémentaires et complexifie le développement des logiciels qui les pilotent.

Une cause supplémentaire de complexité est l'interconnexion de plus en plus grande des différents matériels informatiques : dans certains systèmes d'informations, le nombre d'équipements connectés est fréquemment de l'ordre du millier (en 1997, le système de réservation Amadeus comptait 180 000 équipements répartis dans plus de 100 pays). Cette interconnexion a dépassé le cadre des systèmes d'informations particuliers pour concerner tendanciellement l'ensemble des systèmes informatiques existants avec la naissance et le développement d'Internet. Elle n'a pu s'effectuer que grâce à une transformation architecturale majeure des systèmes informatiques : d'une conception hiérarchique, appelée "maître-esclave", où une machine (gros système ou mainframe qui concentre l'essentiel des ressources

décennie soixante, à six minutes dans la décennie soixante-dix, à quatre secondes dans la décennie quatre-vingt, et à quatre centièmes de seconde dans la décennie quatre-vingt-dix, avec une capacité de transport multipliée par un million sur l'ensemble de la période. Les progrès pourtant spectaculaires que constituent l'apparition du TGV apparaissent minimes par comparaison (1998, p 38).

informatiques), supervisait les actions des autres machines (principalement des terminaux "passifs"), on est passé à une architecture "client-serveur", où toute machine peut selon les situations être traitée en client ou devenir serveur, c'est à dire fournisseur d'un service ou d'une ressource. Le terme serveur fait référence à tout processus qui reçoit une demande de service venant d'un client via un réseau, traite cette demande et renvoie le résultat au demandeur (client). Ce type d'architecture "distribuée" permet le traitement coopératif d'applications, c'est à dire la communication directe de deux applications via un réseau, ou encore la communication directe de deux processus d'une même application répartie sur un réseau, chaque processus s'exécutant sur la machine la plus appropriée. Le modèle client-serveur a permis une rationalisation de l'utilisation des ressources matérielles, en exploitant l'augmentation de la puissance à un coût réduit représentée par les micro-ordinateurs et les stations de travail. Par contre, il a nécessité d'importants efforts en termes de développement des logiciels pour traiter les problèmes de communication entre systèmes, de gestion de bases de données distribuées et de gestion de systèmes distribués, et une refonte des applications existantes, voire de toute l'organisation informatique⁹. La souplesse d'évolution d'une telle architecture, et en conséquence la variété croissante des services proposés, expliquent la rapidité de sa diffusion : aux Etats-Unis le pourcentage des applications sur systèmes client-serveur est passé de 27 % en 1993 à 51 % en 1995 et est estimé à 76 % en 1997 (OCDE, 1997 B, p. 35).

3 - L'objectif d'automatisation de fonctions de plus en plus complexes

Mis à part le calcul scientifique, l'informatisation a concerné au départ des tâches de *back-office* simples et répétitives (gestion de la paye, facturation, opérations bancaires élémentaires...). Si le volume des opérations pouvait être important, ce qui justifiait l'utilisation de l'ordinateur, les logiciels étaient peu complexes (ce qui ne signifie pas que rapportées aux connaissances de l'époque ils aient été simples à mettre au point), d'autant que les tâches à informatiser avaient souvent été préalablement standardisées et taylorisées et que leur traitement informatique s'effectuait en temps différé.

L'intégration des différentes fonctions

⁹ Une architecture client-serveur réduit les dépenses en matériel mais augmente celles en logiciels et en services plus complexes (Eurostat, 1996 A, p. 87).

Une première dimension de la complexification des logiciels provient de deux phénomènes liés : la volonté d'intégrer les différentes procédures –elles-mêmes de complexité croissante - qui avaient été automatisées séparément (gestion de la production, des finances et de la logistique), et d'effectuer ces différentes opérations de façon interactive et en temps réel. Cette volonté d'une informatisation complète du système de gestion de l'entreprise correspond à une évolution de la vision du système d'information de l'entreprise comme étant au cœur de celle-ci. Elle s'est traduit par le développement du marché des logiciels de gestion globale d'entreprise (ERP, *Entreprise Ressources Planning*), des bases de données, de la gestion des réseaux. Cette globalisation de l'informatisation peut avoir en retour des conséquences importantes sur le fonctionnement de l'entreprise : "il ne s'agit plus seulement d'« informatiser » un service ou une activité, mais bien de repenser les modes de production et d'organisation d'une activité, compte tenu des potentialités de l'informatique" (Jacques De Bandt, 1995, p. 84). Les technologies de l'information ne servent plus uniquement à automatiser des tâches et des métiers existants dans l'entreprise, et sont de plus en plus utilisées pour repérer et faciliter l'émergence de nouveaux métiers et services.

Les types de décisions informatisées

La deuxième dimension de la complexification qui s'est opérée en parallèle avec la précédente concerne le type de décisions modélisées dans les logiciels. Les décisions peuvent être classées en trois niveaux : opérationnel, coordination, stratégique (François Pichault, 1990, p. 73). Certes tout processus de décision peut se décomposer en phases d'identification, d'évaluation, de choix et de ratification (Herbert Simon, 1980) et comprend des périodes d'investigation, de transmission et de traitement d'information, mais le type d'informations concernées est différent.

Par exemple, Takayasu Miyakawa (1997, p. 67) distingue quatre modalités d'usage de l'information dans la sphère productive : l'information routinière, l'information mercatique (développement et protection des marchés), l'information opérationnelle (liée à la conception et à la mise en œuvre de la production), l'information managériale (caractère stratégique). De même, Luc Rubiello (1997, p. 76-78) différencie l'information de production, l'information tactique et l'information stratégique. L'information de production (ou opérationnelle) manipule des données de base (brutes) volumineuses en taille et en nombre, mais concerne des opérations dont l'informatisation est ancienne à partir d'un outil de base commun - le système de gestion de bases de données (SGBD) - , qui sont relativement stables et bien

maîtrisées (comptabilité, gestion des stocks, paye, facturation). L'information tactique est obtenue par le traitement des données du niveau précédent et repose sur des outils de pilotage (interrogations avec les langages de quatrième génération L4G, systèmes informatisés d'aide à la décision SIAD). L'information stratégique repose sur des informations issues de la couche tactique, elle correspond à des besoins mal définis et changeant rapidement et s'appuie sur des outils plus récents comme les *Executive Information System* (EIS). Les problèmes décisionnels correspondants sont le contrôle opérationnel (par exemple, le contrôle des stocks) pour l'information de production, le contrôle tactique (par exemple, l'analyse budgétaire) pour l'information tactique, et la planification stratégique (par exemple, la localisation des lieux de production) pour l'information stratégique.

Ces décisions n'ont pas toutes le même niveau de complexité (Eric Brousseau, 1993, p. 224) : certaines peuvent être qualifiées de simples, comme le tri d'une liste par ordre alphabétique, et d'autres de complexes, comme le design d'un nouveau produit. Le premier type de décision est le fruit d'un raisonnement totalement logique (un calcul) et fait partie de ce que Ehad Zuscovitch (1983, p. 51) nomme la partie "algorithmée" de l'entreprise (ou plus généralement d'une organisation). Le deuxième type de décision fait appel à un raisonnement qui n'est pas totalement formalisé (structuré, décomposé en phases analytiques séparées), et/ou repose sur des informations non formalisées. Disposant d'un savoir incomplet sur le monde, la vision du monde des autres agents et l'avenir, le décideur se fie largement à son intuition et à son expérience, fait appel aux résultats de processus d'essais-erreurs observés dans le passé qu'il remet en perspective, par analogie, avec la situation présente, qui n'est pas immédiatement déductible d'une situation antérieure et connue (Eric Brousseau, 1993, p. 224). Ces décisions représentent la partie "non algorithmée" de l'organisation (Ehad Zuscovitch, 1983, p. 51). Naturellement les premières décisions qui ont été informatisées correspondent à la partie algorithmée en commençant par les plus simples et les plus fréquentes (routinières). L'informatisation de ces décisions, en permettant la collecte systématique de données facilement manipulables et stockables sur les opérations réalisées, a ouvert des opportunités pour l'automatisation de décisions plus complexes de caractère plus stratégique. On peut citer les exemples de la gestion logistique à partir de la gestion des stocks, de la conception des produits basée sur l'automatisation de leur fabrication. Cette tendance que Ehad Zuscovitch (1983, p. 52) nomme "algorithmisation de la firme", repose sur la distinction (par rapport à l'incertitude caractéristique de la partie non algorithmée) entre une incertitude réductible (difficultés à assembler toute l'information pour prendre une décision

dans un temps limité) et une incertitude irréductible (provenant de la simultanéité des actions des autres et des autres événements extérieurs) (idem, p. 51). Si certaines activités de décision échappent à toute tentative de formalisation ou de structuration en raison de "l'irréductibilité de l'être humain" (François Pichault, 1990, p. 13), d'autres ne le sont qu'en raison de l'état actuel de nos connaissances. Les conséquences de cette évolution sont le développement de logiciels plus complexes reposant sur des algorithmes eux-mêmes plus complexes ou sur des heuristiques (outils d'aide à la décision).

On peut caractériser cette évolution en utilisant la typologie des technologies immatérielles de R.L Daft et N.B. Mac Intosh (1978) basée sur deux critères : la variété des tâches d'une part, leur degré de précision, de clarté, d'intelligibilité d'autre part. L'informatisation a d'abord concerné les technologies programmables (variété faible, clarté élevée). Elle a concerné ensuite les technologies de savoir-faire (variété et clarté faibles, par exemple cambiste) et les technologies technico-professionnelles (variété et clarté élevées, par exemple la fonction juridique). Enfin elle atteint les technologies de recherche (variété faible, clarté élevée) dont un exemple est la formulation de la stratégie.

L'automatisation des relations

La troisième dimension de la complexification concerne l'utilisation des technologies de l'information et de la communication pour automatiser des relations ou établir de nouvelles relations entre agents économiques. Jusqu'à la fin des années quatre-vingt, l'application des technologies à des tâches et des fonctions étendait l'automatisation à de nouveaux domaines d'applications (activités de bureau, technologies de production flexibles) mais sans modification fondamentale par rapport aux phases précédentes de l'automatisation (Alain Rallet, 1997, p. 96). Une rupture qualitative se produit dans les années quatre-vingt-dix avec l'intégration de l'informatique et des télécommunications, qui va permettre aux technologies de l'information et de la communication de réaliser pleinement leur potentiel de "*mediating technology*" (Claudio Ciborra, 1993) en portant sur les mécanismes de coordination entre les agents économiques, sur deux plans complémentaires. Premièrement à l'intérieur de l'organisation avec le développement de nouvelles techniques de travail de groupe (*groupware*) en relation avec des modifications dans l'organisation des activités, plus centrées sur la réalisation complète d'un produit, la maîtrise globale des processus et des événements, et l'accomplissement collectif d'un projet que sur une stricte séparation des tâches. Deuxièmement, à l'extérieur de l'organisation, l'utilisation des technologies de l'information et

de la communication pour gérer les relations avec les partenaires extérieurs de l'organisation, que celles-ci soient occasionnelles avec des clients (commerce électronique) ou plus stables avec des fournisseurs et sous-traitants (Echange de Données Informatisées, et plus récemment la mise en place *d'extranet*). Le bon déroulement de ces interactions entre des acteurs différents, disposant d'équipements hétérogènes, concernant des informations textuelles mais aussi sonores et visuelles, avec des exigences croissantes en termes de sécurité, de confidentialité, de fiabilité et de convivialité, nécessite la mise au point de logiciels de plus en plus complexes.

Le traitement des connaissances

La quatrième dimension de la complexification concerne l'utilisation des technologies de l'information pour traiter les connaissances. Après les déboires de l'intelligence artificielle dans sa volonté de produire des connaissances, dont les principales réalisations se sont limitées au développement de systèmes experts avec des succès divers selon les domaines, les tentatives actuelles concernent plus modestement la gestion des connaissances, que celles-ci soient internes à l'organisation (*knowledge management*) ou externes (intelligence économique). Cependant si les objectifs sont moins ambitieux, la complexité (dans tous les sens du terme) des outils logiciels nécessaires à ces activités est par contre extrêmement élevée.

4 - Les logiciels comme créateurs de complexité : la complexité proactive

La plus grande partie de la complexité que doit maîtriser un logiciel provient "des nombreuses institutions humaines et les nombreux systèmes auxquels les interfaces du logiciel doivent se conformer" (Frederick P. Brooks, 1996, p. 159). C'est en effet, le logiciel qui doit s'adapter " parce qu'il est le dernier arrivé sur le terrain ou parce qu'il est perçu comme étant le plus facile à rendre compatible au reste" (idem). Mais les logiciels ne sont pas seulement plus complexes parce qu'ils doivent répondre à des situations de complexité croissante. Dans de nombreux cas, ce sont les potentialités offertes par les logiciels qui contribuent à complexifier (parfois considérablement) les problèmes que devront résoudre les logiciels. La *complexité proactive* désigne cette situation où se crée de la complexité par la production de nouveaux problèmes, au-delà de la complexité inhérente à la résolution des problèmes déjà existants.

Michel Crozier (1963) a montré comment les entreprises évoluent en complexifiant leurs règles. Cette tendance, freinée par les capacités limitées des services organisationnels, peut avec les potentialités de l'informatisation s'exprimer pleinement par l'addition de nouvelles réglementations et procédures de plus en plus sophistiquées. Cette complexité proactive concerne également la multiplication des types de produits, de prestations, de cas différents, de situations qu'il faut envisager, et le raccourcissement des délais de réaction. Le traitement automatique de l'information a facilité le développement des politiques de flux tendus et de zéro stocks qui en retour augmentent la complexité des situations à gérer : par exemple, Districast, distributeur de l'éditeur Casterman, a réussi en optimisant sa logistique, à livrer ses clients en 24 heures. De ce fait, ceux-ci réduisent leurs propres stocks, ce qui a eu pour conséquence une augmentation considérable des commandes d'un montant unitaire plus faible, le volume des ventes restant à peu près stable. En réponse à cette complexité supplémentaire, Districast a automatisé la préparation de ces commandes (Catherine Paliarne, 1998, p. 110). Un exemple caricatural de complexité créée en partie artificiellement est fourni par le système Socrate de la SNCF, dont il a du reste fallu réduire la variété des situations possibles pour que le logiciel fonctionne correctement et que le client puisse s'adapter. De nombreux autres exemples peuvent être cités comme le calcul des salaires (multiplicité des situations concernant le temps de travail, les rémunérations de plus en plus individualisées, les prélèvements comme la CSG déductible et non déductible), le nombre de références que gère la grande distribution, les différentes formules de prêt des banques, les modalités d'obtention des diplômes...

C - DES CHANGEMENTS TECHNIQUES PERMANENTS

Sur une période relativement courte, l'économie du logiciel se caractérise par la permanence des changements techniques qui tout à la fois constitue une réponse à la complexité croissante des problèmes à résoudre et permet le développement de cette complexité (complexité proactive). Les principaux changements sont tout d'abord présentés (1). Le fait que ces changements concernent l'ensemble des modèles d'innovations que recense l'économie de l'innovation confirment leur richesse et leur diversité (2). Les conséquences d'une telle profusion d'innovations sont une instabilité technologique permanente (3).

1 - Des changements techniques dans tous les aspects de la production des logiciels...

Les multiples changements techniques concernent aussi bien l'extension accélérée des domaines d'applications des logiciels, la création de nouveaux produits, le renouvellement des méthodes de production, les innovations architecturales, les innovations dans la programmation et l'apparition de nouveaux langages, et l'utilisation d'outils d'automatisation (génie logiciel).

a - L'extension des domaines d'application

L'extension des domaines d'application des logiciels s'effectue schématiquement selon trois dimensions (l'automatisation, la communication et la modélisation) et affecte une partie de plus en plus importante de l'activité économique et sociale, avec une complexité croissante des problèmes que doivent résoudre les logiciels.

Tableau VI

Dimension	Du plus simple au plus complexe		
Automatisation	Application de règles simples	Système-expert, systèmes d'aide à la décision.	Gestion de la relation client
Communication	Présentation, P.A.O.	Multimédia, site Web	<i>Groupware</i> , Commerce électronique
Modélisation	Tableur	C.A.O.	Réalité virtuelle

Concernant l'automatisation, celle-ci a d'abord concerné l'application de règles simples dans les domaines de la gestion et du pilotage de processus¹⁰, puis des tentatives de simuler les comportements et raisonnements effectués par des personnes réelles pour résoudre un problème se rapportant à un domaine d'activité précis (système expert¹¹). Ces systèmes remplissent, avec des succès différenciés selon les domaines, des fonctions de diagnostic, de

¹⁰ Récemment la gestion informatique de processus s'est étendue à la circulation automatique des documents (*workflow*).

¹¹ Certains auteurs préfèrent parler de système à base de connaissances, laissant la qualité d'expert à la personne qui est détentrice d'un ensemble homogène de connaissances (Luc Rubiello, 1997, p. 17).

conception, de prévision ou d'optimisation. Plus récemment c'est dans le domaine de la gestion de la relation client que se multiplient les efforts d'automatisation : recueil automatique de multiples informations sur les clients, établissement de relations significatives entre ces critères et propositions personnalisées (en termes de produits, voire en termes de prix différenciés).

Concernant la communication, l'informatisation a permis d'améliorer la présentation d'informations de plus en plus variées (texte, image, plan, schéma...) et l'intégration d'informations situées sur les supports les plus divers (multimédia). Elle vise maintenant à faciliter les relations de coopération entre membres d'une même organisation (*groupware*¹²) et de coordination entre différentes entreprises (places de marché électroniques pour le commerce entre entreprises notamment).

Concernant la modélisation, l'informatique est devenue un outil courant dans la représentation de situations simples (à l'aide d'un tableur par exemple). Elle est utilisée pour la création d'objets virtuels (biens mais aussi services) qui constituent une aide précieuse dans les activités de conception. Un pas supplémentaire est franchi avec l'utilisation d'une "réalité virtuelle" dans des situations problématiques (simulateur de vol, explosion nucléaire...) ou à des fins esthétiques (effets spéciaux dans le cinéma par exemple).

Bien évidemment un même domaine d'application peut connaître une informatisation qui concerne les trois dimensions : par exemple, la gestion informatisée des stocks comporte des aspects d'automatisation (actualisation des stocks en fonction des achats et des ventes), de communication (visualisation de l'état des stocks, mécanismes d'alerte en deçà d'un certain seuil) et de modélisation (optimisation du niveau des stocks en fonction des prévisions). De même, la plupart des logiciels ont des fonctions qui couvrent ces trois dimensions : un banal traitement de texte est un outil de communication, qui dispose de fonctions automatisées (mise en forme, vérification orthographique et grammaticale, résumé, table des matières) et qui permet une certaine modélisation de la pensée de l'auteur (vision instantanée des effets d'une modification, mode "plan", liens hypertextes..).

¹² Le *groupware* est l'ensemble des méthodes, procédures, logiciels et plates-formes informatiques permettant à des personnes, associées dans un même contexte professionnel, de travailler ensemble avec le maximum d'efficacité.

b - La création de nouveaux produits

L'extension des domaines d'applications suscite la création permanente de nouveaux types de logiciels. Par exemple, la gestion automatisée de la relation clients suppose la constitution de gigantesques entrepôts de données (*data warehouse*) et la création d'outils logiciels (*data mining*) permettant de les exploiter efficacement. Le développement du *groupware* nécessite le développement de logiciels (parfois appelés collecticiels) et l'adjonction de fonctions spécifiques à des logiciels bureautiques permettant à un réseau de personnes bien identifiées d'échanger des informations et de créer collectivement des documents en groupe et en temps réel. La création de places virtuelles d'échanges entre entreprises nécessite la mise en place d'applications partagées entre plusieurs entreprises (*crossware*). Le développement de fonctions plus complexes voit la mise en place d'agents logiciels "intelligents", spécialisés dans un certain nombre de tâches (par exemple la recherche d'informations pertinentes) et capables de se perfectionner par apprentissage, leur rôle étant d'interpréter, de façon toujours plus pertinente, les intentions de l'opérateur et d'exécuter, en large autonomie, des séquences complexes d'opérations¹³.

La création de nouveaux domaines d'applications nécessite également le développement d'outils qui ont une utilité indirecte. Par exemple le développement des réseaux a suscité l'apparition de logiciels de cryptographie, de compression de données, de sécurité (*firewalls*) et la constitution d'annuaires électroniques universels (par exemple *Novell Directory System*) qui permettent de gérer automatiquement les profils des utilisateurs (avec les autorisations correspondantes), les adresses électroniques, les chemins d'accès...

Le développement de nouveaux produits peut aussi consister en des améliorations significatives des produits existants. Par exemple, la plupart des Systèmes de Gestion de Bases de Données qui étaient à l'origine "hiérarchiques" se sont transformés en SGBD "relationnels" pour pouvoir gérer la plus grande complexité des relations existant entre les données. L'utilisation des différentes applications a été grandement facilitée par l'introduction d'une interface graphique, appelée WIMP (*Windows, Icons, Menus, Pointer*), que Frederick P. Brooks considère comme "une des innovations logicielles les plus remarquables de ces vingt dernières années" (1996, p. 227).

¹³ Par exemple pour accomplir des tâches comptables, ont été développés des "robots comptables" qui sont des agents pouvant exécuter automatiquement des procédures logicielles complexes dans la comptabilité.

Enfin, l'apparition de nouveaux produits peut résulter de l'intégration dans un seul produit, de fonctions auparavant effectuées par des produits distincts (progiciels intégrés de gestion d'entreprise, suite bureautique intégrée).

c - Le renouvellement des méthodes de production des logiciels

La question des méthodes de production des logiciels est peut-être le domaine qui a connu les changements les plus rapides suivant une évolution assez semblable à celle de l'ensemble de l'économie sur une longue période¹⁴. D'une quasi-absence de méthode formalisée caractéristique d'une production de type artisanal, la production de logiciels a vu l'adoption de méthodes de plus en plus formalisées avec un découpage de plus en plus fin en phases séparées que l'on peut rapprocher des principes de l'Organisation Scientifique du Travail. La complexité croissante du dessin des lettres qui représentent les différentes phases de ces méthodes (méthode en V, puis en Y et enfin en W) témoigne de l'augmentation du nombre d'étapes qui s'exécutent séquentiellement. Dans ce type de méthodes, les relations entre les développeurs et les utilisateurs (ou ceux qui les représentent) se limitent aux phases les plus en amont (définition des besoins et spécifications du logiciel) et les plus en aval (validation du logiciel). L'extrême importance des phases amont, où toute imperfection entraîne des coûts de correction ou de modification très élevés, a conduit à lui consacrer des méthodes spécifiques : par exemple, la méthode JAD (*Joint Application Design*) élaborée par IBM en 1977, qui consiste à réunir et à faire travailler ensemble les utilisateurs et les informaticiens avec comme objectif des décisions prises en commun et des spécifications acceptées par l'ensemble des participants.

Malgré ces efforts, l'allongement des processus de développement et l'insatisfaction terminale fréquente des utilisateurs¹⁵ a conduit à l'adoption de *méthodes itératives*, permettant des interactions permanentes entre concepteurs et utilisateurs et dont l'objectif est une adaptation plus rapide et plus précise aux besoins. Un premier pas dans cette direction a été la modification du "modèle de la cascade" - dénommé ainsi parce qu'à chaque étape il était

¹⁴ La *Software Productivity Research Inc* identifie 65 méthodes différentes de développement de logiciel (Warnier-Orr, Yourdon, Jackson, Merise, RAD...).

¹⁵ "Ce n'est qu'après avoir achevé la construction du système qu'on peut découvrir qu'il est trop malcommode à utiliser, qu'il a des performances inacceptables, ou qu'il est dangereusement exposé aux erreurs ou aux stratagèmes des utilisateurs (...). L'examen des spécifications est censé découvrir ce genre de problème au début du processus, mais le test d'utilisation réel est irremplaçable" (Frederick P. Brooks, 1996, p. 231).

seulement possible de descendre à l'étape suivante - par Winton Royce (1970) en incluant une rétroaction d'une étape vers la précédente et une limitation de cette rétroaction à l'étape qui précède immédiatement, de façon à limiter les coûts et les délais qu'elle entraîne (Frederick P. Brooks, 1996, p. 230). Un modèle qui approfondit cet objectif est le modèle du développement "en spirale" de Barry W. Boehm, qui introduit la notion de cycle de développement itératif, avec des possibilités de retour et de modification à tous les niveaux et pas seulement à la phase immédiatement précédente. L'aboutissement de cette évolution est constitué par les méthodes évolutives où les phases de spécification (quoi faire ?) et de conception (comment faire ?) sont étroitement imbriquées. Ces méthodes, moins formalisées, reposent sur le prototypage rapide (un prototype est utilisé pour clarifier les besoins des utilisateurs en leur permettant de réagir au vu du système¹⁶, et pour définir le système final), et le prototypage évolutif (ou développement incrémental) où, à partir d'une première version rapidement développée et aux fonctionnalités limitées, sont créées des versions successives utilisables, par l'intégration d'éléments nouveaux (complexification), la suppression d'éléments inutiles (simplification), la prise en compte de nouveaux aspects du problème (divergence), et l'intégration d'éléments du système restés déconnectés (convergence). Outre les possibilités de rectification en cours de développement et la motivation que constitue la vision permanente de la concrétisation des efforts accomplis, l'intérêt de ces méthodes est qu'elles permettent une répartition du pouvoir plus équilibré entre les concepteurs et les utilisateurs, qui "forment une équipe de résolution de problèmes" (Robert Michon, Lin Gingras, 1988, p. 129) et qu'elles facilitent l'intercompréhension mutuelle, grâce à l'existence de versions précoces qui créent "un pont sémantique entre concepteur et utilisateur" (idem, p. 130).

d - Les innovations dans la programmation et l'apparition de nouveaux langages

La programmation, malgré sa jeunesse, a connu des innovations importantes qui se sont le plus souvent traduites par l'apparition de nouveaux langages de programmation, un langage

¹⁶ Les documents produits lors des premières phases de nombreuses méthodes, par exemple les modèles conceptuels de données de la méthode Merise, sont souvent trop complexes et trop abstraits pour l'utilisateur (Robert Michon, Lin Gingras, 1988, p. 127). Il est difficile d'exprimer des besoins précis sans voir le système, comme il est difficile de spécifier sans apprécier les problèmes de réalisation (Jean-Marc Geib, 1989, p. 17). Face à ces difficultés, "Le but d'un prototype est de rendre tangible la structure conceptuelle spécifiée, de telle manière que le client puisse en tester la cohérence et l'ergonomie" (Frederick P. Brooks, 1996, p. 172).

de programmation n'étant rien d'autre qu'un ensemble déterminé et fixe d'instructions qui permettent la transcription du problème considéré.

La première grande innovation a été l'introduction de la *programmation structurée*, au début des années soixante dix, avec notamment le langage Pascal, qui visait à éviter les instructions de branchement (le célèbre Go To dont la nocivité avait été mise en évidence dans un article célèbre d'Edsger Dijkstra en novembre 1968) grâce à des structures conditionnelles (si... alors...sinon) et itératives (les boucles pour, jusqu'à, tant que). L'objectif est d'avoir un schéma de raisonnement prédéfini et des entrées et sorties prédéfinies, et de conduire les opérations de façon rationnelle et prédéterminée, par opposition aux méthodes d'improvisation. En forçant l'exposition de la structure du raisonnement suivie jusque dans le code source du programme, la programmation structurée permet de contrôler la rigueur des raisonnements et doit rendre plus facile la compréhension et la modification des programmes.

La deuxième innovation qui prolonge la précédente a été l'introduction de la *programmation modulaire* en découpant un programme en une arborescence hiérarchisée de multiples modules et sous modules. C'est une conception fonctionnelle descendante, dans laquelle on s'efforce de minimiser les communications et les contrôles entre modules, et de maximiser la cohésion interne de chaque module individuel. On peut y voir une illustration de la théorie des systèmes complexes de Herbert A. Simon (1974), où celui-ci montre l'efficacité résultant d'une décomposition hiérarchique et de l'utilisation de formes intermédiaires stables.

La troisième innovation est la naissance de la *programmation orientée objets*, apparue avec le langage Smalltalk, qui est basée sur les notions de *classe*, d'*héritage* et d'*encapsulation*. A l'opposé de la programmation classique qui sépare données et procédures, l'approche orientée-objet supprime cette dichotomie en intégrant données et fonctions de traitement (appelées méthodes) dans des entités appelées classes. Une *classe* est définie par une structure, des propriétés et un ensemble de méthodes utilisées pour manipuler les objets qui sont les instances de cette classe. L'*encapsulation* est le fait de masquer la structure interne et les traitements d'un objet (partie privée), l'accès à cet objet ne pouvant être effectué que par l'intermédiaire de messages activant les méthodes définies dans son interface publique. L'*héritage* est le fait qu'une classe peut être définie par extension ou spécialisation d'une autre classe (héritage simple) ou de plusieurs autres (héritage multiple). L'intérêt principal de la programmation objet est qu'elle permet une meilleure maîtrise de la complexité des systèmes, qu'elle facilite la réutilisation des composants logiciels déjà développés et qu'elle permet le

développement rapide d'applications, notamment en utilisant les méthodes évolutives précédemment exposées.

Enfin, on peut mentionner plus récemment la conception de "logiciel modulaire" (OCDE, 1997 B, p. 185) qui vise à reproduire les avantages de la programmation orientée objet en les rendant accessibles aux utilisateurs comme aux programmeurs. L'idée est de développer des éléments modulaires comme les *applets* Java de Sun, les composants ActiveX de Microsoft ou OpenDoc d'Apple, qui sont téléchargés lorsqu'ils sont utilisés (donc mis à jour automatiquement), qui peuvent être associés et assortis sur n'importe quel type de matériel parce qu'ils partagent un ensemble d'interfaces standard. L'utilisation de ces éléments de petite taille, qui remplissent chacun une fonction limitée mais qui interagissent étroitement avec d'autres éléments, doit permettre d'adapter le logiciel aux besoins précis de l'utilisateur à un moment donné. La généralisation de ce nouveau modèle à de nombreux domaines dépendra de façon décisive de la réussite du processus de standardisation des interfaces de ces composants.

e - L'utilisation d'outils d'automatisation : le génie logiciel

La production de logiciels a été analysée comme une succession de processus de codifications. L'allongement de ce processus par la création d'étapes intermédiaires lui confère une plus grande efficacité (cf. chapitre I), les étapes les plus en aval pouvant être réalisées automatiquement (par exemple un compilateur qui traduit automatiquement un programme écrit en langage de haut niveau¹⁷ en langage machine). L'idéal, impossible à atteindre, serait de pouvoir programmer directement en langage naturel, des "outils" réalisant automatiquement la conversion en langage machine. De ce point de vue, on peut dire que "la programmation automatique a toujours été un euphémisme désignant la programmation à l'aide d'un langage de plus haut niveau que ce dont dispose le programmeur" (Frederick P. Brooks, 1996, p. 167). L'efficacité de l'allongement du détour de production est un phénomène habituel dans la production économique, mais qui est amplifié par une particularité des outils logiciels : à la différence d'un outil matériel comme une machine, sa reproduction a un coût négligeable, une fois qu'il a été mis au point.

¹⁷ Un langage de plus haut niveau est un langage qui a un vocabulaire plus étendu, une syntaxe plus compliquée, et une sémantique plus riche. Le langage de plus bas niveau est le langage machine, le langage de plus haut niveau est le langage naturel, les différents langages de programmation occupant des positions intermédiaires entre ces deux extrêmes.

De ce fait, les efforts pour développer des outils logiciels de plus en plus sophistiqués sont constants depuis les débuts de la programmation. Ils se sont amplifiés avec la naissance du génie logiciel, "domaine controversé s'il en est, qui après avoir fait beaucoup rêver dans les années 1980, a aussi beaucoup déçu, avec des gains de productivité qui n'étaient pas au rendez-vous, et des échecs industriels patents" (Jacques Printz, 1998, p. 232). Le génie logiciel, terme apparu en 1968 à l'initiative du Département de la Défense des Etats-Unis, est un ensemble de méthodes et de techniques qui visent à convertir l'activité de réalisation de logiciels, d'un artisanat mystérieux et faillible en une discipline d'ingénierie (OCDE, 1991, A, p. 13). Il comprend l'ensemble des activités de conception et de mise en œuvre des produits et des procédures tendant à rationaliser la production du logiciel et son suivi. Un de ses objectifs est d'accentuer l'automatisation de la production des logiciels, "l'ambition avouée [étant] de faire disparaître les programmeurs au profit des générateurs de codes, ce que proposent déjà certains ateliers de génie logiciel" (Patrick Jaulent, 1992, p. 29)¹⁸.

On peut distinguer dans le génie logiciel deux dimensions. La première est une démarche mathématique et scientifique, très structurée, qui à partir de l'utilisation de méthodes formelles vise à éliminer les "mauvaises pratiques" génératrices d'erreurs et d'ambiguïtés. Un exemple en est la preuve formelle de programmes pour démontrer mathématiquement qu'un programme est correct. La deuxième dimension, plus pragmatique, consiste en en une activité pratique de développement d'outils de programmation visant à accroître la productivité des réalisateurs de logiciels (Génie Logiciel Assisté par Ordinateur).

Sont ainsi apparus au début des années quatre-vingt, une grande variété d'outils logiciels ou outils CASE (Computer-Aided Software Engineering) qui couvrent l'ensemble des phases de la production des logiciels : analyse des besoins, conception, estimation du coût et du travail, gestion de projet et documentation, production automatique de programmes et essai, vérification et validation des systèmes. Au sein de ces outils, on distingue les outils *Lower Case*, centrés sur les activités de programmation et de test, des outils *Upper Case* qui correspondent aux phases amont d'analyse et de conception.

¹⁸ La recherche d'une automatisation croissante de la production est, dans le cas des logiciels, particulièrement recherchée en raison des qualifications élevées des informaticiens, et de leur relative pénurie face à l'importance de la demande, qui expliquent des rémunérations élevées.

Au-delà de l'utilisation d'outils isolés, une tentative plus ambitieuse consiste à fournir un ensemble intégré d'outils et de méthodes, ou I-CASE (*integrated CASE*), suite d'outils CASE homogènes et cohérents qui permettent de traiter tous les aspects d'un projet (phases du cycle de vie, gestion de projet, analyse), et dont le plus connu est AD-Cycle d'IBM.

2 - ...qui correspondent à l'ensemble des modèles d'innovations de certaines théories évolutionnistes.

A partir de l'étude de l'innovation dans les services, Faïz Gallouj et Olivier Weinstein (1997) ont élaboré une typologie des innovations, dont un des intérêts est qu'elle permet de mettre en évidence la richesse des innovations dans les services, que des approches plus "technologistes" ne perçoivent pas. Cette typologie est construite à partir d'une représentation d'un service comme un système de caractéristiques et de compétences, représentation que nous avons mobilisée pour analyser les caractéristiques des logiciels (cf. chapitre I). *Or quand on examine les innovations dans le domaine des logiciels, on constate qu'elles concernent tous les types d'innovations mis en évidence par Faïz Gallouj et Olivier Weinstein, ce qui témoigne de la forte dynamique innovatrice de l'économie du logiciel.*

Elle est illustrée tout d'abord par la fréquence des *innovations radicales*, qui concernent la création d'un produit totalement nouveau (par exemple le tableur ou plus récemment le navigateur), la création d'une architecture fondamentalement différente (par exemple le client-serveur) ou d'une nouvelle façon de programmer (par exemple la programmation orientée objet). Mais les innovations radicales ne constituent qu'une petite partie des innovations, les autres innovations (innovations incrémentales) étant différenciées en plusieurs catégories dans la typologie de Faïz Gallouj et Olivier Weinstein.

La première catégorie concerne les *innovations d'amélioration* qui consiste simplement à améliorer certaines caractéristiques. Dans le cas des logiciels, ce type d'innovations est fréquent et important : par exemple, la possibilité de gérer un beaucoup plus grand nombre d'enregistrements dans le cas d'un système de gestion de bases de données, ou la forte augmentation du nombre de couleurs et de niveaux de détail que peut traiter un logiciel de retouche d'images ou de photographies.

La deuxième catégorie concerne les *innovations "incrémentielles"* qui s'opèrent par transformation ou adjonction de caractéristiques. Pour les logiciels, un exemple spectaculaire de transformation d'une caractéristique importante (la facilité d'utilisation) est l'introduction

d'une interface graphique. L'adjonction de caractéristiques est réalisée par l'introduction de nouvelles fonctionnalités à chaque version (fréquente) d'un logiciel, que facilite la nature particulière d'un logiciel.

La troisième catégorie, appelée *innovation de recombinaison*, consiste à jouer sur les possibilités de combinaisons multiples de différentes caractéristiques. Une illustration est la création d'un produit logiciel unique en intégrant différents produits existants (progiciel intégré bureautique ou de gestion d'entreprise). Plus récemment, la conception de logiciels modulaires, exposée précédemment, est significative des possibilités et des potentialités de ce type d'innovations pour les logiciels.

La quatrième catégorie est représentée par des *innovations ad hoc*, qui est définie comme la construction sociale interactive d'une solution au problème particulier d'un client donné, souvent coproduite et reconnue a posteriori, non reproductible en tant que telle puisqu'elle est localisée et contextualisée, mais qui peut servir pour d'autres prestations. Ce type d'innovations concerne plus le développement de logiciels sur mesure, où des méthodes et des schémas de résolutions de problèmes originaux peuvent être réutilisés. Avec le développement de la programmation orientée objet, ce type d'innovations peut s'étendre à des composants logiciels développés.

3 - Quelques conséquences de cette profusion d'innovations : une instabilité technologique permanente

Il faut distinguer le rythme soutenu d'apparition d'innovations, de leur processus de diffusion qui peut être lent pour plusieurs raisons. Tout d'abord, il faut tenir compte du temps nécessaire pour assimiler une innovation et effectuer l'apprentissage de son utilisation¹⁹. De plus, l'apprentissage a un caractère ambivalent : plus il est performant dans une direction, plus il est freiné dans les autres directions (Olivier Favereau, 1998, p. 199). De ce point de vue, une volonté de changements trop rapides peut être contre-productive. Serge Bouchy souligne que "l'intégration perpétuelle de nouvelles techniques sur lesquelles l'état des connaissances

¹⁹ Capers Jones cite une étude non publiée d'IBM qui montre que "le transfert de technologies est un processus terriblement lent". Cette étude concluait que la diffusion d'une technologie est une fonction du temps analogue à une suite de Fibonacci : une nouvelle méthode met un an pour toucher 15 %, trois ans pour 50 % et cinq ans pour monter à 90 % (1989, p. 276). Ce processus peut être beaucoup plus long pour une innovation majeure : il s'est écoulé vingt ans entre les premiers langages orientés objets dans les laboratoires et l'apparition de produits commerciaux effectivement utilisés.

n'en est parfois qu'aux balbutiements, (...) freine l'assimilation et le réel transfert des connaissances" (1994, p. 143). De même Jacques Printz estime que la frénésie innovatrice à laquelle on assiste depuis le début des années quatre-vingt-dix est un véritable obstacle à une croissance en quantité et en qualité des logiciels" (1998, p. 323)²⁰.

L'importance des effets d'apprentissage peut expliquer pourquoi des innovations porteuses de solutions à long terme ne sont pas adaptées en raison de leurs coûts immédiats importants. Ces coûts comprennent le coût de l'innovation elle-même²¹, et les coûts de migration et de formation, qui peuvent être élevés quand l'innovation représente une discontinuité dans le développement technologique : par exemple, l'adoption du génie logiciel représente un changement de paradigme dans les méthodes de réalisation du logiciel avec des techniques, des compétences, des méthodes, des outils et des modes de travail radicalement nouveaux (OCDE, 1991 A, p. 30)²².

Dans certains cas, des rigidités sociales et institutionnelles peuvent également entraver l'introduction d'innovations. Il faut également prendre en compte, notamment quand il existe des incompatibilités entre technologies, la crainte d'effectuer un mauvais choix, d'introduire une innovation qui sera rapidement dépassée par une autre. Paradoxalement, "lorsqu'une évolution rapide est escomptée, cette expectative elle-même ralentit l'application des nouvelles techniques ; dans un monde en évolution rapide, les gens tendent à adopter une politique attentiste" (Arnold Heertje, 1988, p.11). Or dans l'économie du logiciel, et plus généralement dans l'informatique, il est difficile de faire des pronostics sur les évolutions même à un horizon de court/moyen terme.

²⁰ F.G. Roux ironise sur "les objets autres nouveautés miraculeuses comme les langages de quatrième génération, les ateliers de génie logiciel, l'intelligence artificielle, les systèmes experts ; la réponse unanime des responsables informatiques est que « l'avenir de l'informatique passe par l'orientation objet qui est la seule solution qui sera porteuse de productivité et de qualité » ; il y a 10 ans, ils disaient que ce serait Merise, 8 ans les AGL et l'IA, 5 ans Unix; 3 ans le modèle client-serveur" (L'informatique professionnelle n° 135, Juin Juillet 1995, p.30-31).

²¹ Par exemple les très hauts niveaux d'investissement nécessaires expliquent pourquoi les petites organisations productrices de logiciels n'utilisent pas les ateliers de génie logiciel, y compris certaines sociétés qui ont participé à la mise au point de ces AGL (OCDE, 1991 A, p. 26).

²² Emmanuel Saint-James souligne que l'habitude de la communauté informatique d'introduire chaque nouvelle idée par un langage nouveau, oblige "à se mettre dans la tête une multitude d'idiosyncrasies arbitraires et à réécrire les bibliothèques de programmes" (1993, p. 4).

Il en résulte une très grande hétérogénéité technologique avec des processus de diffusions qui peuvent avoir des rythmes très différents selon les acteurs, et une économie du logiciel qui semble condamnée à une instabilité technologique permanente : "la rapidité de développement de technologies nouvelles empêche constamment toute velléité de stabiliser les choix technologiques. Le cycle expérimentation puis rationalisation et stabilisation est ainsi constamment réactivé par l'offre dont les effets d'annonce fonctionnent d'autant plus comme des injonctions d'achat qu'une forte incertitude règne sur l'environnement technologique futur" (Alain Rallet, 1997, p. 86).

Une illustration est la profusion des langages informatiques utilisés : par exemple, alors que de multiples tentatives ont existé pour promouvoir un langage universel soutenu par des acteurs puissants (PL/1 par IBM à l'époque où elle dominait l'informatique mondiale, puis ADA par le Département de la Défense des Etats-Unis avec un investissement qui a dépassé le milliard de dollars, JAVA actuellement par Sun), la diversité des langages utilisés n'a cessé de s'accroître, et un langage comme Cobol reste très utilisé alors que "depuis le début des années soixante-dix, au moins, on annonce pour prochaine sa fin" (Jean Bres, 1994, p. 5)²³.

Nous avons vu que le logiciel était un objet particulièrement difficile à appréhender sur un plan technique. Son analyse sur un plan économique est au moins aussi délicate, en raison des difficultés pour lui appliquer les catégories économiques habituelles.

Section II - L'"insaisissabilité" théorique : les difficultés à appliquer les catégories économiques habituelles

En tant que produit, le logiciel est difficilement appréhendable par les catégories économiques habituelles : il occupe une position intermédiaire entre les biens et les services (A), il possède certains attributs des biens collectifs (B), il est difficile de lui appliquer les

²³ Cet auteur cite une étude de Micro-focus qui estime qu'il existe dans le monde 100 milliards de lignes de code, et que sur ces 100 milliards, 77 % sont écrites en Cobol.

théories de la valeur qui ont été forgées par les économistes pour expliquer l'évolution des prix (C).

A - UN PRODUIT HETEROGENE OCCUPANT DES POSITIONS DIVERSES SUR LE CONTINUUM BIENS-SERVICES

L'hétérogénéité traitée dans ce paragraphe concerne la variété des formes technico-économiques des logiciels et non la diversité des fonctions des logiciels analysée précédemment. Sous cet aspect, les logiciels peuvent se positionner à différents endroits du continuum existant entre les biens et les services. Après avoir analysé les tendances à une fusion entre les biens et les services (1), nous verrons la possibilité et la nécessité de maintenir une distinction entre les biens et les services, en intégrant une différenciation entre les biens tangibles et les biens intangibles (2), que nous utiliserons pour caractériser les différentes formes de logiciels (3).

1 - Vers une fusion des biens et des services ?

Certaines analyses ont mis en évidence une quasi-disparition des frontières entre les biens et les services, à partir du constat de la régression des services purs (nécessitant peu d'appuis matériels) et des biens purs (n'intégrant pas de services). Ces analyses s'appuient sur l'existence de deux tendances : la tendance à l'industrialisation des services avec le développement de processus "d'objectivation" et de standardisation des services, liés notamment à leur informatisation, dont une forme extrême est la production de quasi-biens (Faïz Gallouj, 1997, p.30) ; la tendance à la "servicialisation" des biens comportant une part croissante de services et intégrant une relation de service dans une volonté de répondre plus finement aux besoins du client (production sur mesure). Certains auteurs infèrent de ces tendances l'absence de distinction fondamentale entre les biens et les services : par exemple, André Barcet, Joël Bonamy, et Anne Mayère (1984, p. 128) considèrent la notion de "produit-service" comme une unique catégorie économique pouvant revêtir différentes formes ; C. Belleflamme, J. Houard, B. Michaux et O. Ruysen (1986) rejettent eux aussi l'opposition bien/service en considérant que ce qui intéresse le client c'est la satisfaction d'un besoin (d'une fonction) ; pour ces auteurs, chaque produit est le résultat, dans des proportions variables, de la combinaison d'un processus de production et d'un processus de servuction (différenciation et adaptation du produit aux spécifications de l'utilisateur).

Sans nier l'intérêt de ces analyses et la réalité des tendances évoquées, il nous semble néanmoins possible et nécessaire de maintenir une distinction entre les biens et les services²⁴. Notre position peut se résumer ainsi : à certains égards, cette distinction n'est pas pertinente, mais pour analyser certaines questions, elle le demeure.

2 - Une distinction maintenue entre des biens tangibles, des biens intangibles et des services

Nous reprenons ici l'analyse de Jean Gadrey (1999 A). Tout en rejetant les oppositions classiques et contestables entre les biens et les services fondées sur les critères de matérialité, de stockabilité, de périssabilité, de transportabilité, Jean Gadrey réexamine dans un premier temps la différence de nature existant entre les biens et les services, énoncée par Peter Hill (1997) : seuls les biens sont des entités existant indépendamment de leurs producteurs et de leurs consommateurs ; sur les biens peuvent être établis des droits de propriété ; à l'inverse les services ne sont pas des entités, ce sont des changements de condition ou d'état d'une entité possédée par le demandeur du service, ce qui a comme conséquence que le produit d'une activité de service ne peut être revendu indépendamment de l'entité sur laquelle il porte. A partir de cette opposition, Jean Gadrey effectue deux ajouts importants. Premièrement, la notion d'entité, qui joue un rôle central dans la distinction opérée par Peter Hill entre les biens et les services, mais qui n'est pas vraiment définie, peut être appréhendée à partir de l'existence d'une identité sociale et historique des biens, attestée par la présence de "marques identitaires", en précisant que le processus d'attribution d'une identité à un bien n'est pas simplement technique mais de nature conventionnelle. Deuxièmement, la définition des services comme changement de condition ou transformation d'état d'une entité ne permet pas d'intégrer une partie importante de ce qui est considéré usuellement comme des services (restauration, hôtellerie, spectacles, une partie de l'activité commerciale...). Pour englober l'ensemble du champ des services, Jean Gadrey propose de centrer la définition des services sur la mise à disposition temporaire d'une capacité technique et humaine d'une organisation prestataire à un client-usager, pour produire des effets utiles sur lui-même ou sur des biens qu'il possède.

²⁴ "En dépit de ces formes de convergence entre les deux modèles, avec un secteur industriel devenant de plus en plus intensif en service à la fois sur le plan de ses process et sur celui de ses produits, il semble prématuré de conclure à une similarité des situations ; (...) la dimension de service des produits industriels reste dans la majorité des cas de type plus périphérique que centrale, et la tendance à l'individualisation est loin d'être répandue" (Jean Gadrey, 1996 A, p. 193).

Dans le même article, Peter Hill introduit une distinction importante au sein des biens, entre les biens tangibles et les biens intangibles. Les biens intangibles sont définis comme étant les "originaux" produits d'une activité de création scientifique, technique ou artistique. Ces "originaux" inscrits sur des supports divers peuvent être facilement dupliqués à un coût de plus en plus faible et généralement sans commune mesure avec le coût de création de l'original. Les biens intangibles se limitent aux cas où "l'original et les copies ont les mêmes caractéristiques utiles" (Jean Gadrey, 1999 A, p. 16). Dans certaines activités de création artistique (peinture, sculpture...) où l'original a une valeur bien supérieure à la copie, les originaux peuvent être considérés comme des biens tangibles non reproductibles. Les biens intangibles sont des biens dans la mesure où ce sont des entités sur laquelle des droits de propriété peuvent être établis, même si ces droits de propriété prennent fréquemment des formes spécifiques (copyright, droit d'auteur) en raison de certaines caractéristiques des biens collectifs que possèdent les biens intangibles. Ces biens sont intangibles en raison de leur immatérialité, de leur absence de dimensions et de coordonnées physiques, ou plus précisément de l'absence d'importance sur un plan économique de leur localisation géographique. Les biens intangibles occupent une place de plus en plus importante dans l'activité économique au niveau de la production (descriptifs d'un produit, d'un composant, d'une méthode de production) et de la consommation finale (texte, film, composition musicale...).

3 - L'application à l'informatique et la situation particulière des logiciels

Ces distinctions sont particulièrement fécondes pour analyser les différentes formes technico-économiques que peuvent prendre les programmes informatiques. Dans des cas extrêmes, le code-objet d'un programme informatique peut être gravé dans le silicium et être intégré de façon indissociable au composant électronique sur lequel il agit. Ces composants électroniques programmés sont des biens tangibles. Il s'agit certes d'une activité où le coût de conception du premier exemplaire est particulièrement élevé mais où la production d'un exemplaire supplémentaire n'est pas une opération triviale. Mais, et c'est la situation la plus fréquente, le code-objet d'un programme informatique peut exister également indépendamment des composants électroniques sur lesquels il agit, être fixé sur un support matériel distinct (cartes perforées, disques magnétiques ou opto-électroniques...), ce qui a

pour conséquence qu'il devra être chargé dans une mémoire (mémoire vive²⁵) pour pouvoir être exécuté par les composants électroniques pour lesquels il a été conçu.

Logiciel et progiciel

Il importe dans cette situation de distinguer deux possibilités très différentes. Une première possibilité est le développement par une entreprise (ou par le service informatique interne de l'organisation) d'un logiciel sur mesure en réponse à un besoin précis. Dans ce cas, la production de logiciel peut être considérée comme une activité de services, le prestataire mettant ses compétences à la disposition d'un client-usager pour développer un programme informatique. Les spécificités de ce produit unique et original sont négociées entre le prestataire et le client, l'adéquation aux besoins dépendant étroitement de la qualité de la relation de service qui s'instaure entre eux. La deuxième possibilité est la production préalable d'un logiciel standard, dont les producteurs escomptent qu'il répondra aux besoins d'utilisateurs anonymes, et qui est vendu "sur catalogue". Il s'agit ici typiquement de la production d'un bien intangible, la production d'un nouvel exemplaire strictement identique à l'original étant une simple opération de copie à un coût dérisoire. Cette forme de logiciel est appelée progiciel (par contraction de produit-logiciel), défini comme étant un "ensemble complet et documenté de programmes, conçu pour être fourni à plusieurs utilisateurs en vue d'une même fonction". (Journal Officiel du 7 décembre 1980). Une définition plus développée indique qu'il s'agit d' "un ensemble cohérent et indépendant, comprenant des programmes, des services, des supports de manipulation de l'information conçus pour réaliser des traitements informatiques standard, proposé suivant une forme commerciale et qu'un usager peut utiliser de façon autonome après une mise en place et une formation limitée" (01 Informatique, 24 mai 1983). On peut remarquer que la production de logiciels sur mesure est constitutive d'une logique de demande (qui doit s'exprimer avant le démarrage de l'activité), alors que la production de progiciel repose sur une logique d'offre (mise sur le marché d'une production déjà réalisée) (Christine Babelon, 1987, p. 88).

²⁵ En informatique, on distingue fondamentalement deux types de mémoires, même si se développent de plus en plus des mémoires "intermédiaires" entre ces deux catégories : les mémoires mortes ou ROM (Read Only Memory) dont le contenu est figé définitivement lors de la production du composant, et les mémoires vives ou RAM (Random Access Memory) dans lesquelles on peut stocker de façon temporaire des programmes, des données... (le contenu est perdu lors de la coupure de l'alimentation électrique).

Le tableau suivant fait apparaître le positionnement différencié des logiciels selon leur nature technico-économique.

Tableau VII : Les logiciels entre les biens et les services.

Biens tangibles		Biens intangibles		Services		
Autres biens	Composants électroniques programmés	Progiciels		Logiciels sur mesure	Autres services	
		Outils et systèmes	Applicatifs	Autres services informatiques		Services informatisés
		Produits culturels multimédias				
		Autres biens intangibles				

Cette représentation appelle deux remarques complémentaires. Premièrement, au-delà des relations de complémentarité entre les composants matériels et logiciels, il existe des possibilités de substitution entre les différentes formes que peut prendre un même programme informatique. Ce constat illustre la remarque plus générale de Michel Callon (1993, p. 25-26) selon lequel "il n'existe en réalité aucune barrière infranchissable entre un énoncé, une machine ou un corps discipliné, les substitutions et les réinscriptions [étant] toujours possibles, au moins dans une certaine mesure". Un exemple historique célèbre est le calcul des logarithmes, que Charles Babbage voulait faciliter en le décomposant en différentes opérations dont les plus simples pourraient être réalisées automatiquement par une machine qu'il était en train de concevoir. Cette tâche a été effectuée par des logiciels dès l'apparition des premiers ordinateurs, elle est maintenant intégrée dans des composants électroniques présents sur la moindre calculatrice de poche. Pour prendre un exemple actuel, si l'on considère le secteur des jeux informatiques, secteur particulièrement typique de cette diversité, on constate qu'un même jeu informatique peut être réalisé de multiples façons. Ce jeu peut reposer uniquement sur des composants électroniques (machine dédiée), comporter

une partie matérielle (la console de jeu et son système d'exploitation) et une partie progicielle (le jeu lui-même), fonctionner sur un micro-ordinateur où le matériel se caractérise par son universalité, le système d'exploitation et le jeu étant réalisés sous forme de progiciels. Mais ce même jeu peut également avoir été écrit spécifiquement par l'auteur-utilisateur sur un micro-ordinateur comportant un système d'exploitation standard, voire dans le cas des premiers micro-ordinateurs vendus en kit sans aucun programme existant, avoir nécessité le développement par l'utilisateur d'un système d'exploitation puis du jeu lui-même.

Deuxièmement, les frontières entre les différentes catégories ne sont pas étanches et le rattachement d'une production concrète à une activité n'est pas toujours évident, ce qui pose des problèmes d'interprétation des statistiques existantes.

Entre le logiciel sur mesure et le progiciel, il existe des produits intermédiaires dérivés des produits standards : progiciels paramétrables pour être adaptés aux spécificités de l'entreprise cliente, modules vendus séparément et fournis dans le cadre d'une prestation intégrant des services. La STERIA, qui la première en France a développé ce type d'activités, parle de semi-produits pour désigner ces solutions constituées d'un noyau commun et d'une partie sur mesure, dont elle estimait qu'elle revenait deux fois moins chère qu'une solution intégralement sur mesure (Eric Labat, 1984, p. 248). De ce fait, il est parfois difficile de délimiter le champ statistique des progiciels de celui des logiciels sur mesure, d'autant que face aux reproches adressés aux SSII françaises de ne pas faire de progiciels, certaines ont tendance à rebaptiser à des fins commerciales "progiciels" la simple existence de "quelques briques de base", l'essentiel étant pourtant écrit sur mesure (Gérard Dréan, 1996 A, p. 278).

Les services informatiques

Parmi les services liés à l'informatique, il est nécessaire d'opérer une distinction entre les services informatisés, qui avaient une existence antérieure à leur informatisation, même si celle-ci a pu transformer plus ou moins profondément la nature de l'activité, par exemple la comptabilité, et les services informatiques qui sont des services de support à l'activité informatique.

Les services de support comprennent d'une part des services basés sur des engagements contractuels de moyens (conseil, assistance, formation) et d'autre part des services reposant théoriquement sur des engagements de résultats (développement de logiciel, ingénierie des systèmes et intégration de systèmes). L'ingénierie de systèmes se distingue du développement

de logiciels par la fourniture d'un système complet comprenant du matériel. Elle est appelée intégration de systèmes si le matériel comprend des éléments hétérogènes en provenance de constructeurs différents. Les services informatiques constituent un "monde varié et instable, parallèle à celui des produits, auquel il est lié par des relations complexes de complémentarité et de concurrence" (Gérard Dréan, 1996 A, p. 260). L'évolution est rapide avec l'apparition et la disparition périodiques de certaines catégories de services. Par exemple, la tierce maintenance applicative (confier à un prestataire la montée en charge et l'évolution d'une nouvelle application), apparue récemment, se développe rapidement. En sens inverse, les prestations matérielles ou prestations-machine (traitement à façon, télétraitement par lots à distance, télétraitement en temps réel, vente d'heures machines) qui représentaient l'essentiel des services informatiques au début de l'informatique ont quasiment disparu au profit de prestations intellectuelles (Christine Babelon, 1987, p. 68-69)²⁶. Ces prestations matérielles constituaient le "service bureau" où les travaux des clients étaient traités sur les matériels du prestataire. Le service-bureau résultait de l'équipement en ordinateurs du service bureau classique qui existait depuis plusieurs décennies. Avec l'équipement progressif en matériel informatique des entreprises, le service-bureau a cessé d'exister entraînant la disparition ou la reconversion des premières sociétés de service qui s'étaient créées à cette occasion. Il faut toutefois noter que le récent développement du "*facilities management*", consistant pour une entreprise à faire sous-traiter son informatique à une société de service, redonne une certaine jeunesse peut-être temporaire à ce type d'activités (cf. chapitre VI). Dans l'ensemble des services informatiques, le développement de logiciels occupe une place importante, mais qui n'est pas toujours évidente à mesurer séparément.

Des problèmes de délimitation

Parmi les progiciels, on peut distinguer les progiciels outils et systèmes, des progiciels applicatifs. Ce sont les logiciels outils et systèmes qui, de par leur plus grande proximité avec le matériel et ses producteurs, ont été le plus rapidement et le plus massivement fournis sous forme de progiciels (phénomène de standardisation), les logiciels d'applications n'étant

²⁶ Le traitement de données qui représente environ 15 % des services informatiques, en baisse régulière au rythme de 1 à 2 % par an, est actuellement essentiellement constitué de services d'application informatisés comme la gestion des transactions par cartes de paiement, ou les services aux experts comptables ou aux agents d'assurances. Ces prestations figurent dans les services informatiques lorsqu'ils sont fournis par des sociétés considérées comme des sociétés de services informatiques, alors que des prestations comparables n'y figurent pas lorsqu'elles sont fournies par d'autres entreprises. Selon Gérard Dréan, "la présence du traitement dans les services en informatique est essentiellement un archaïsme statistique" (1996 A, p. 268).

concernés que plus récemment (notamment avec l'explosion de la microinformatique) et restant plus proches des producteurs de service. La séparation entre composants matériels et composants logiciels n'est pas toujours évidente, notamment avec l'apparition de la microprogrammation qui a introduit un niveau intermédiaire (le microcode) entre les fonctions directement réalisées par les circuits électroniques et celles réalisées par programmation : un microprocesseur contient actuellement une quantité importante de logiciels. De ce fait, la distinction traditionnelle entre matériel et logiciel devient de plus en plus conventionnelle (Gérard Dréan, 1996 A, p. 21). Enfin, l'importance et la pertinence de ces délimitations sont perçues différemment selon le type d'acteurs concerné. Pour un utilisateur final non informaticien, ce qui importe ce sont les services fournis globalement par le système informatique et il lui est indifférent de savoir ce qui dans ces services dépend de dispositifs matériels ou logiciels. Pour un développeur d'applications la question importante est de connaître les fonctionnalités offertes globalement par la "couche" constituée du matériel et de son système d'exploitation, la répartition entre ce qui relève du matériel ou du logiciel de base n'ayant qu'une importance secondaire pour l'optimisation éventuelle des performances de l'application. Par contre, cette séparation reste importante et fait l'objet d'arbitrages technico-économiques délicats pour ceux qui conçoivent l'architecture du matériel d'une part, l'écriture du système d'exploitation d'autre part. Enfin, pour l'économiste, le critère principal concerne la nature économique du produit avec d'un côté des biens tangibles (composants) ou intangibles (logiciels) dont la production s'apparente à une production industrielle, et d'un autre côté des services (logiciels sur mesure ou autres services informatiques).

Le dernier problème de délimitation, qui prend une importance grandissante avec la banalisation de l'utilisation de l'informatique dans les domaines les plus divers, est la séparation, dans un produit, entre ce qui relève d'une activité de programmation (la composante logicielle du produit ou de l'activité) et ce qui relève de la mobilisation de compétences non informatiques. C'est particulièrement net avec le développement de produits multimédias. Par exemple, doit-on considérer que l'encyclopédie Universalis sur CD-ROM est un logiciel en insistant sur les nouvelles fonctionnalités de recherche et de navigation (liens hypermédias) permises par ce type de support, ou considérer qu'il s'agit toujours d'un livre en insistant sur la permanence du contenu par rapport à sa version papier ? De même, la réalisation d'un site Web mêle inextricablement des compétences de graphistes, de communicateurs et d'écritures de programmes informatiques. La fusion des activités de

programmation et des activités non informatiques que réalisent les programmes, est une tendance inéluctable liée à l'éloignement des logiciels du matériel et à la réalisation de fonctions de plus en plus sophistiquées par l'informatique.

B - LE PROGICEL, UN BIEN PARTIELLEMENT COLLECTIF ?

Quand un programme informatique existe sous forme de progiciel, se posent des problèmes de production et de diffusion, spécifiques aux biens intangibles, qui possèdent certaines caractéristiques des biens collectifs. Si ces questions ne concernent a priori que les progiciels et non les logiciels sur mesure, il importe de remarquer que les progiciels constituent une part croissante des logiciels, que les frontières entre ces deux activités ne sont pas étanches et que de plus en plus des logiciels sur mesure seront produits à partir de composants logiciels standardisés, pour lesquels on rencontre les mêmes types de problèmes que pour les progiciels. Après avoir constaté que les progiciels possèdent "naturellement" les déterminants des biens collectifs (1), nous analyserons les mécanismes techniques (2) et juridiques (3) élaborés pour contrôler sa reproduction et en faire un produit commercialisable.

1 - Les caractéristiques des biens collectifs et les progiciels

Les biens collectifs ont des caractéristiques particulières concernant leur production (indivisibilité), leur utilisation (bien non rival) et leur distribution/acquisition (non-exclusion de l'usage).

a - Indivisibilité

Un bien est indivisible quand le coût de production est indépendant du nombre d'utilisateurs. Or un progiciel possède une très grande facilité de reproduction : "créer un nouvel exemplaire à partir d'un programme existant est rapide, peu coûteux et ne nécessite pas de matériel spécialisé puisque le matériel destiné à utiliser le programme permet aussi de le reproduire" (Gérard Dréan, 1996 A, p. 219)²⁷. De ce fait, on peut dire que le progiciel se caractérise par une indivisibilité croissante, liée à la baisse du coût des supports sur lesquels il est fourni et dont l'importance a toujours été négligeable par rapport au coût de

²⁷ Pamela Samuelson, experte en droit informatique, fait remarquer que vendre un logiciel ressemble à vendre une voiture avec les clés de l'usine. Lauren Ruth Wiener, qui rapporte cette observation, ajoute qu'il est plus simple de copier une disquette que de faire tourner une usine automobile (1994, p. 105) !

développement de l'original (baisse du coût des disquettes et CD, documentation sous une forme numérique au détriment de la forme papier...). Pour le producteur, la vente d'une nouvelle unité a même un coût nul quand le progiciel est téléchargé sur Internet, situation qui se développe rapidement. De ce point de vue, on peut dire que la production d'un progiciel s'apparente à la production de connaissances de type scientifique et technique (coûts de production très lourds et coûts de reproduction quasiment nuls).

On peut toutefois nuancer légèrement cette caractérisation, si l'on ne considère pas uniquement le nombre des utilisateurs, mais également leur diversité et les spécificités des matériels qu'ils utilisent. En effet le coût de développement du progiciel original croît significativement quand ce progiciel est destiné à être utilisé par des personnes aux compétences différentes, sur des matériels de plus en plus hétérogènes et en interaction avec d'autres progiciels très divers. Enfin, il est évident que le coût des services de maintenance, d'installation, de formation, parfois fournis avec le progiciel, augmente avec le nombre de progiciels livrés, mais on peut considérer que ces services sont distincts sur le plan de l'analyse économique du progiciel lui-même, comme le prouve le fait qu'ils peuvent être assurés par d'autres prestataires que la société productrice du progiciel.

b - Bien non-rival

Un bien non rival est un bien qui ne se détruit pas dans l'usage et qui peut donc être adopté par un nombre infini d'utilisateurs, sa consommation ou son usage par un agent économique n'empêchant pas sa consommation ou son usage par les autres (indivisibilité dans l'usage). Si l'on considère que ce qui constitue fondamentalement un progiciel c'est "l'original" du produit, il est clair qu'il s'agit d'un bien non-rival. C'est moins évident si l'on prend en compte comme identité économiquement pertinente, un exemplaire particulier inscrit sur un support donné et en général identifié par un numéro de série. En effet, son utilisation par un nouvel agent économique présuppose une opération de copie. Mais outre la facilité de cette opération, le fait que le détenteur du progiciel qui a servi à la copie le détienne toujours après la réalisation de la copie, fait que celle-ci apparaît fréquemment comme légitime, à défaut d'être toujours légale. Par conséquent, il n'existe une spoliation (limitée) que par rapport au

créateur de l'original, ce qui fait qu'elle est considérée très différemment, d'un point de vue éthique, de l'appropriation d'un bien (tangible) appartenant à autrui²⁸.

c - La non-exclusion de l'usage

Un bien est appropriable s'il est possible pour celui qui l'utilise ou le consomme d'exclure tout autre utilisateur ou consommateur potentiel. Cette propriété d'*excludability*, que Michel Callon (1994 A, p. 6) propose de traduire par "exclusivisme", n'existe pas a priori pour un logiciel, dont il est difficile d'empêcher l'utilisation par un agent économique. Les progiciels, comme les autres biens intangibles, se caractérisent par leur fluidité. Leur production n'est pas limitée par des contraintes physiques, leur consommation n'est pas soustractive et l'on peut parler à leur propos d'économie d'abondance (Charles Goldfinger, 1994, p.14-15).

Toutefois, il existe des possibilités techniques ou juridiques de rendre un progiciel appropriable, permettant de le transformer en un produit commercialisable. Il est intéressant de noter que les efforts²⁹ portent sur le rétablissement artificiel d'une situation de rareté au contraire de l'activité économique "normale" qui est de lutter contre celle-ci³⁰, selon une logique qui n'est pas sans rappeler les tentatives de création de semences qui ne peuvent se reproduire.

2 - Les restrictions du caractère collectif par des procédés techniques

Deux solutions existent sur un plan technique : la transformation en un bien tangible et l'adjonction d'un dispositif technique empêchant la reproduction du progiciel sans l'accord de la société productrice.

²⁸ Un constat identique peut être fait pour le photocopiage des livres et plus généralement pour la reproduction "sauvage" des biens intangibles.

²⁹ En analysant de façon plus globale les dimensions de bien public de la connaissance, Sandra Braman note que "les efforts déployés pour trouver des moyens diversifiés de transformer cette connaissance en propriété privée représentent un des plus âpres combats de cette décennie" (1997, p. 99).

³⁰ "Un produit vendable se trouve dans une situation de rareté relative. Le logiciel est-il une ressource rare ? En un certain sens oui, mais cette rareté s'entretient artificiellement par des mécanismes juridiques. Elle n'est pas naturelle" (Bernard Lang, 1998 C).

a - Par la transformation en biens tangibles

La première solution, la plus radicale, consiste à graver le programme dans le silicium sous forme d'un composant électronique ROM. Il s'agit dès lors d'un bien tangible qui ne possède plus aucune caractéristique des biens collectifs. Cette solution qu'avait envisagé Bill Gates lors de la production de son premier progiciel (un interpréteur BASIC pour micro-ordinateur) face à la recopie des cartes perforées sur lequel il était stocké³¹, constitue une protection efficace contre la copie, dans la mesure où celle-ci nécessite des compétences, du matériel spécialisé et est coûteuse³², mais elle présente beaucoup d'inconvénients qui font qu'elle est peu utilisée en pratique. Tout d'abord, elle est difficilement applicable pour des programmes de taille importante. Ensuite, même quand il est produit en grande série la production d'une unité supplémentaire a un coût non négligeable, ce qui rendrait ce programme peu compétitif par rapport à un programme concurrent commercialisé sous forme de progiciel. Enfin et surtout, à la différence d'un logiciel qui peut être modifié très facilement, un composant électronique se caractérise par sa rigidité ce qui pose deux problèmes majeurs : premièrement, la correction des défauts initiaux, inévitables dès qu'un programme atteint un certain niveau de complexité, nécessite de détruire les composants défectueux et d'en produire de nouveaux, alors qu'il est aisé de distribuer une nouvelle version voire un simple additif logiciel (*patche*) dans le cas d'un progiciel³³. Deuxièmement, il est plus difficile de produire et de faire accepter des nouvelles versions d'un composant électronique que d'un progiciel, dans un secteur où l'obsolescence programmée est une source de revenus importante.

b - Par l'adjonction de dispositifs techniques

La deuxième solution consiste à adjoindre un dispositif technique visant à empêcher la copie du progiciel (le dispositif est présent sur le support de stockage du progiciel) ou à lui

³¹ Du reste, certains micro-ordinateurs commercialisés au début des années quatre-vingt ont comporté un interpréteur BASIC stocké dans la mémoire ROM.

³² Cette efficacité est toutefois relative à l'évolution des techniques : la reproduction d'un CD-ROM, opération difficile et coûteuse il y a quelques années, est devenue un jeu d'enfant (dans tous les sens du terme) actuellement.

³³ Une illustration de cette différence est fournie par les conséquences économiques importantes pour la société productrice (Intel) du constat exceptionnel d'un défaut dans la programmation du microprocesseur Pentium II, comparativement à l'existence habituelle de défauts dans la plupart des progiciels commercialisés qui n'affectent nullement les résultats des sociétés productrices.

retirer tout intérêt (le progiciel ne peut fonctionner qu'avec la présence sur le matériel d'un composant difficilement reproductible livré avec le progiciel). L'efficacité de ces dispositifs est limitée : développement de logiciels permettant de copier des disquettes même protégées³⁴, modification du progiciel pour contourner la vérification de la présence du composant électronique. En effet à la différence de dispositifs de protections voisins comme les décodeurs de télévision, il n'est pas forcément nécessaire de reproduire le dispositif matériel qui peut toujours être neutralisé par une opération logicielle. Certes cette activité peut être très complexe mais elle a toujours suscité une inventivité débordante de la part de *hackers* opposés à la rétention de la circulation d'une information technique (Philippe Breton, 1990, p. 111). Leur motivation réside principalement dans la réalisation d'un exploit technique (d'où l'inefficacité de la sophistication des protections techniques)³⁵, les progiciels "déprotégés" étant rarement revendus, voire même fréquemment non utilisés par ceux qui ont effectué l'opération. Par contre leur diffusion est rapide et il n'est pas rare de voir apparaître une version "déprotégée" avant même la commercialisation du progiciel considéré, phénomène qui ne peut que s'étendre avec le développement d'Internet³⁶.

Par contre ces formes de protection peuvent rendre le progiciel moins efficace, être la source de complications et surtout constituer des difficultés d'utilisation (pouvant aller jusqu'à l'impossibilité d'utilisation) pour des utilisateurs "légaux". C'est pour ces raisons que ces dispositifs sont de moins en moins présents sauf pour certains progiciels particulièrement coûteux et destinés à un public spécialisé.

Aucune solution de nature purement technique ne répondant correctement aux exigences de protection des progiciels (Jean-Benoît Zimmermann, 1995 B, p. 182), c'est sur le

³⁴ Ces logiciels sont même, dans certains cas, commercialisés tout à fait légalement, sous le prétexte de permettre de réaliser une copie de sauvegarde.

³⁵ Une association célèbre de *crackeurs* ("pirates informatiques") est l'association d00dz (*day zero*) dont l'objectif est de forcer les sécurités des logiciels de jeu, le jour même de leur sortie officielle.

³⁶ Le même phénomène se reproduit actuellement pour les œuvres musicales où le format MP3 de transfert audio sur Internet permet une reproduction facile des œuvres. Des tentatives existent pour promouvoir d'autres standards et les logiciels correspondants, de façon à limiter les copies (en nombre et dans le temps) mais ces tentatives sont régulièrement détournées par des *hackers* : le dernier exemple est le logiciel multimédia de Microsoft, Windows Media Audio 4.0, qui devait permettre d'imposer le respect des droits d'auteur en limitant la gravure à un CD sur une période limitée (la copie s'altérant automatiquement au bout d'un certain temps) ; ses protections ont réussi à être contournées deux jours après l'annonce de son lancement (01 Informatique, 20/08/1998).

terrain juridique que vont porter la plupart des efforts pour garantir "l'appropriabilité" des logiciels.

3 - Les restrictions du caractère collectif par des procédés juridiques et la multiplicité des situations existantes

a - L'importance de la question des droits de propriété intellectuelle dans l'économie du logiciel

L'ampleur du "piratage" ou copie illicite de logiciel montre l'importance de cette question. Les Business Software Alliance (BSA) calcule un taux de piratage qui est le rapport entre les logiciels d'application installés (demande) et les expéditions licites de logiciels d'application (offre).

Tableau VIII

***Average Software Piracy Rates by Region, 1994–1998
(percentage of applications installed each year)***

Region	1994	1995	1996	1997	1998
Asia/Pacific	68	64	55	52	49
North America	32	27	28	28	26
United States	31	26	27	27	25
Western Europe	52	49	43	39	37
Latin America	78	76	68	62	60
Eastern Europe	85	83	80	77	76
Middle East/Africa	80	78	74	65	62
World total	49	46	43	40	38

Source : Business Software Alliance, 1999, p. 22.

Il importe toutefois de noter qu'il ne s'agit que d'une estimation dont la fiabilité, comme pour toute mesure d'un phénomène illégal, est difficile à apprécier³⁷. De plus elle émane d'un regroupement des grands acteurs informatiques américains, dont l'intérêt est de souligner l'ampleur du phénomène pour susciter une répression accrue. Dans son rapport de 1999, la BSA estime que le piratage informatique aurait ainsi coûté 109.000 emplois à un secteur qui compte 809.000 salariés et aurait représenté un manque à gagner fiscal de près de 1 milliard de dollars pour l'administration américaine (Business Software Alliance, 1999). La BSA calcule également des pertes de chiffre d'affaires dues au piratage des logiciels :

Tableau IX
Pertes de chiffre d'affaires dues au piratage des logiciels
en milliards de dollars

	1994	1996
Europe	3,9	3,4
Asie / Pacifique	3,1	3,7
Amérique du Nord	3,9	2,7
Amérique Latine	1,0	0,9
Moyen-Orient / Afrique	0,4	0,5

Source :BSA/SPA Global Software Piracy Report, Facts and Figures, 1994-96

Indépendamment de l'importance réelle du piratage, le manque à gagner, estimé par le prix de vente des logiciels piratés, est vraisemblablement surestimé, si l'on considère que de nombreuses sociétés n'auraient pas acheté ces logiciels si elles avaient été contraintes de se les procurer légalement. Il faut de plus ajouter que le piratage d'un progiciel peut aider à la diffusion de sa version commerciale, ce qu'illustre le constat que les logiciels les plus copiés sont également les plus vendus ou les stratégies de fourniture gratuite de certains progiciels pour les imposer sur le marché.

³⁷ Certains indices indirects sont toutefois éloquentes : en Russie un livre sur le logiciel Clipper s'est vendu à 1 193 000 exemplaires alors que Nantucket, l'éditeur de ce logiciel, n'avait enregistré que trois commandes de son produit (Frédéric Dromby, 1999, p. 645).

La répression du piratage a tendance à se durcir et est même devenue un sujet de conflit entre certains pays (par exemple entre les Etats-Unis et la Chine où le taux de piratage est estimé à 90 %), ce qui pourrait expliquer la baisse constatée des taux de piratage et du manque à gagner pour les éditeurs. On peut citer l'exemple d'une agence de publicité brésilienne Artplan, poursuivie par la *Business Software Alliance* (BSA), qui a été condamnée à verser une amende de 65 millions de dollars pour l'utilisation frauduleuse de 382 programmes, soit 170 000 dollars le logiciel (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 132). Par contre, la vente de progiciels en ligne par Internet pose de nouveaux problèmes : le Software & Information Industry Association (SIIA) a effectué une étude entre le 15 et le 20 août sur les logiciels vendus aux enchères sur Internet par de grands sites (eBay, ZDNet.com et Excite.com) d'où il ressort que 60 % étaient des logiciels pirates ; l'étude cite l'exemple de Macromedia Director, un logiciel très utilisé par les graphistes, dont le prix de vente est de 999 dollars, qui était vendu aux enchères au prix de 28 dollars (Les News, 9/9/99).

En plus des problèmes de piratage se pose le problème de l'imitation ("clonage"), qui n'est certes pas propre au progiciel, mais qui a pris une dimension importante dont témoigne le nombre de procès³⁸ et la vigueur des actions entreprises : en 1990, des *hackers* ont été arrêtés par le FBI pour avoir divulgué des extraits du code source du système d'exploitation d'Apple.

Plus généralement pour le secteur du logiciel, "les droits de propriété intellectuelle apparaissent comme une variable clé au moins aussi importante que la capacité productive ou le facteur travail" (Robert Merges, 1996, p. 285). Cet auteur montre comment cette question peut avoir une importance décisive dans la stratégie de l'entreprise et cite l'exemple de l'acquisition d'Asthon-Tate (éditeur de dBase, produit leader à l'époque dans les systèmes de gestion de bases de données pour micro-ordinateurs) par Borland en 1991 ; le Département de la Justice autorisa l'opération à la condition que Borland renonce à des droits de propriété "forts" sur dBase et abandonne un procès contre Fox (un autre éditeur de SGBD). En quelques années Microsoft, qui avait racheté Fox, devint hégémonique sur ce segment de marché avec un produit Access "compatible" avec dBase.

³⁸ Par exemple, Xerox a intenté des procès contre tous les éditeurs de logiciels qui développent des interfaces graphiques pour lesquelles elle s'estime pionnière ; pour les mêmes raisons Apple a attaqué Hewlett-Packard et Microsoft.

Mais si les questions de droits de propriété sont importantes dans ce secteur, elles sont également très complexes en raison de la nature particulière du logiciel.

b - Une question complexe vu la nature particulière du logiciel

La question du statut juridique du progiciel est particulièrement délicate car elle mêle indissociablement deux problèmes. Le premier problème, qui n'est pas spécifique au progiciel, concerne le degré socialement optimal de la protection. Il correspond à un dilemme, qualifié de *schumpeterien*, concernant l'influence du régime des droits de propriété sur le dynamisme technologique, la protection devant tout à la fois constituer un facteur incitatif suffisant *ex ante* sans trop entraver l'imitation et la diffusion *ex post* (Jean-Alain Héraud, 1995, p. 91). De ce fait, le débat a toujours été vif entre partisans d'une protection "forte", qui permet d'accroître les incitations à innover, et partisans d'une protection "faible" insistant sur les bienfaits d'une diffusion facilitée des innovations pour élargir la concurrence et éviter les pertes d'efficacité liées aux effets de duplication de la recherche (Partha Dasgupta, Joseph Stiglitz, 1980).

A ce premier problème s'ajoute la question de la nature particulière du logiciel comme texte numérique actif (cf. chapitre 1), le fait que "le logiciel est à la fois technologie et expression" (Jean-Benoît Zimmermann, 1995 B, p. 182).

En tant qu'expression intellectuelle, un logiciel, est une prestation intellectuelle, assimilable à une œuvre de l'esprit au même titre qu'une œuvre littéraire ou artistique, propriété incorporelle exclusive de son auteur, et semble devoir relever de la législation sur les droits d'auteur, le support matériel sur lequel repose le logiciel n'étant considéré que comme un accessoire à la prestation intellectuelle.

Mais un logiciel est également une technologie, "un processus (...) pour réaliser une tâche de calcul, de traitement et de manipulation de données, de commandes" (Jean-Benoît Zimmermann, 1995 B, p. 183), un procédé permettant de tirer parti des ressources du matériel en vue d'un résultat déterminé. A ce titre, il peut également prétendre à une protection par le brevet d'invention qui permet l'exercice d'un droit de monopole sur un dispositif ou une méthode, consenti selon les critères de nouveauté, d'originalité et de non-évidence. En définitive, le logiciel ne peut "trouver commodément sa place ni dans l'une, ni dans l'autre de ces deux branches du droit de la propriété intellectuelle, ce qui fait toute la difficulté en la matière" (André Lucas, 1987, p. 195). Ces difficultés sont une illustration particulière de "la

crise qui frappe aujourd'hui les systèmes de droit de propriété intellectuelle" (Dominique Foray, 1995, p. 136), le logiciel étant "un « cas d'espèce » qui comporte une signification prospective pour l'ensemble de l'industrie" (Jean-Benoît Zimmermann, 1995 B, p. 182).

Dans le cas du logiciel, il faut de plus prendre en compte les aspects très différents qu'il intègre, et qui ne disposent pas tous du même type et du même niveau de protection : la résolution non informatique du problème (pour un logiciel applicatif) - qui fréquemment reposait en partie sur des savoir-faire, mais qui en étant transformée en connaissances codifiées devient facilement diffusable -, les algorithmes informatiques utilisés, leur mise en application concrète pour résoudre le problème considéré, le *look and feel*³⁹ de l'application, les interfaces avec le matériel et les autres programmes.

D'une façon générale, après d'intenses débats dans les années soixante-dix, les différents Etats ont renoncé à adopter un cadre juridique spécifique pour le logiciel compte tenu de l'ampleur et de la difficulté de la tâche, et les cadres existants (droits d'auteur et brevets) ont simplement été adaptés (Jean-Benoît Zimmermann, 1995 B, p. 188). Au départ, c'est essentiellement le droit d'auteur qui a été mobilisé, avec des modalités différentes selon les pays, mais de plus en plus se développent des tendances à la brevetabilité des logiciels.

c - Des solutions différentes dans le temps et dans l'espace

Pendant longtemps en France, les logiciels n'étaient pas considérés par la jurisprudence comme relevant de la loi sur la propriété des œuvres littéraires ou artistiques (loi du 11 mars 1957). Ce n'est que le 2 novembre 1982 qu'un jugement de la quatrième chambre de la cour d'appel de Paris a accordé un "droit d'auteur" à un informaticien pour un programme d'ordinateur qu'il avait conçu. A la suite d'une série d'arrêts contradictoires, le législateur a adopté une loi (3 juillet 1985) qui étend explicitement à la protection des programmes informatiques le champ de la loi du 11 mars 1957. Cette loi sera complétée par la loi du 10 mai 1994 qui parle pour la première fois de logiciel et plus seulement de programme d'ordinateur. Ces lois excluent clairement de la protection par droit d'auteur les principes et algorithmes, conformément à la jurisprudence française sur la propriété intellectuelle qui précise que les œuvres protégées doivent être cristallisées dans une forme, les idées n'étant pas protégées par le droit d'auteur.

³⁹ Le *look and feel* est, pour l'utilisateur, l'aspect externe du programme, défini notamment par l'interface-utilisateur, la présentation des commandes...

Sur la question de la brevetabilité des logiciels il s'est produit une certaine évolution. La loi sur les brevets d'invention du 2 janvier 1968 exclut explicitement les "programmes ou séries d'instructions pour le déroulement des opérations d'une machine calculatrice" (article 7). Cette position était motivée essentiellement par la crainte d'asseoir sur le long terme la position dominante d'IBM (Jean-Benoît Zimmermann, 1995 B, p. 190). Cette restriction sera confirmée par la loi du 13 juillet 1978 qui visait à harmoniser la législation française avec le texte de la Convention de Munich. Toutefois la jurisprudence très stricte au départ a admis comme brevetable les logiciels présentés comme parties intégrantes et inséparables de processus industriels : "un programme informatique est indirectement brevetable aux conditions de fond suivantes ; il doit constituer une étape d'élaboration d'un procédé industriel brevetable ; il doit satisfaire lui-même aux trois critères de brevetabilité : la nouveauté, le résultat industriel et l'activité inventive" (Cour d'Appel de Paris, affaire Schlumberger, 15-6-81). A l'heure actuelle, ces questions relèvent de plus en plus de la juridiction européenne et c'est au niveau de la Communauté Européenne qu'est débattue la brevetabilité des logiciels.

En Europe

Au niveau européen, une Directive votée par le parlement le 17 avril 1991, adoptée à l'unanimité par le Conseil des Ministres les 12 et 14 mai, est entrée en application pour tous les pays membres le premier janvier 1993. Cette directive définit un droit du logiciel fondé sur la protection du contenu d'un logiciel par le droit d'auteur. Elle exclut du champ de la protection la logique, les algorithmes et la langue de programmation. Elle protège "l'écriture" d'un programme tout en laissant la liberté à d'autres auteurs d'écrire un logiciel différent pour arriver à un même résultat. Elle autorise uniquement la copie de sauvegarde, la correction d'erreurs dans le cadre de l'utilisation prévue et, sous certaines conditions, la décompilation dans une perspective d'interopérabilité des programmes entre eux et avec les ordinateurs (si les éditeurs ne communiquent pas les informations nécessaires à la réalisation d'interfaces).

Par ailleurs, l'article 52-2 c de la Convention de Munich (1973) interdit de breveter un logiciel. Toutefois, après avoir exclu par principe les logiciels de la brevetabilité, l'Office Européen des Brevets a nuancé sa position : en se basant sur le fait qu'un brevet d'invention est délivré pour toute invention nouvelle, impliquant une activité inventive et susceptible d'application industrielle, les chambres de recours de l'Office Européen des Brevets ont considéré un logiciel comme une invention "s'il engendre un effet technique qui va au-delà des interactions physiques normales existant entre un programme d'ordinateur et un

ordinateur". "Si l'objet revendiqué apporte une contribution de caractère technique, la brevetabilité ne devrait pas être mise en cause pour la simple raison qu'un programme d'ordinateur est impliqué" (directive donnée aux examinateurs de l'OEB). En théorie c'est le procédé impliquant le logiciel qui est brevetable, pas le logiciel en tant que tel. En fait, selon Pierre Breese, avocat conseil en propriété industrielle, il suffit de faire passer le logiciel comme "procédé" pour que le brevet soit accepté. Pierre Breese estime qu'il y a aujourd'hui 10 000 brevets européens portant objectivement sur des logiciels (ZDNet, 23-6-99). A l'occasion de la révision de la Convention de Munich, les éditeurs de progiciels ont tenté d'élargir encore les possibilités de breveter des logiciels, ce qui a suscité de vigoureuses oppositions notamment des partisans des logiciels libres. La conférence intergouvernementale européenne du 25 juin 1999 a repoussé à plus tard la décision et a mis en place un groupe de travail chargé de "clarifier la situation" (ZDNet, 30-6-99).

Au Japon

Au Japon la protection de la propriété intellectuelle est en général assez faible. C'est sous la pression des poursuites contre des entreprises japonaises pour violation de droits de propriété relatifs à des logiciels⁴⁰, qu'une intégration du logiciel à la loi sur le copyright sera effectuée en 1985. La nouvelle loi japonaise exclut toutefois de la protection par copyright les langages de programmation, les algorithmes et les interfaces (*look and feel*), et elle maintient le droit à la modification des programmes pour en améliorer l'efficacité ou pour les adapter à un système non compatible. La décompilation est autorisée sauf pour les logiciels américains à la suite d'un accord bilatéral avec les Etats-Unis (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p.74). Il existe toutefois peu de jurisprudence et beaucoup d'incertitudes qui font que la protection par copyright connaît des limites significatives (Robert Merges, 1996, p. 275).

Par contre le système japonais de brevets qui combine droits de propriété faibles, exigences de nouveauté peu élevée et contrainte forte de divulgation précoce (Dominique Foray, 1995, p. 142) se traduit par un nombre élevé de dépôts de brevets sur des procédés liés au logiciel, estimé à environ 35 000 chaque année (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p.141).

⁴⁰ L'affaire la plus célèbre est le procès intenté par IBM à Hitachi et Fujitsu pour utilisation illicite de son système d'exploitation grand système MVS/XA.

Aux Etats-Unis

Aux Etats-Unis, suite aux travaux d'une commission mandatée par le Congrès, la CONTU (*National Commission on New Technolgical Uses of Copyrighted Works*), le logiciel a été pris en compte dans le champ juridique du copyright (*Computer Software Copyright Act* de 1980 qui amendait le *Copyright Act* de 1976). L'application de cette décision à certains aspects particuliers des logiciels a posé des problèmes juridiques complexes. Le *look and feel* d'un logiciel n'a pas de statut juridique précis mais la jurisprudence tend à le faire bénéficier de la protection par copyright depuis que Lotus a obtenu gain de cause contre Paperback en juin 1989 (Jean-Benoît Zimmermann, 1995 B, p. 194). Les interfaces d'un logiciel ont fait l'objet d'interprétations contradictoires mais il semble que se dégage un consensus pour les exclure du droit d'auteur (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 76). Les opérations de décompilation, y compris pour produire des logiciels compatibles, ont été considérées comme une violation du droit d'auteur dans un procès célèbre et qui avaient mobilisé de nombreux acteurs de l'informatique (Sega contre Accolade en avril 1992) et sont interdites à de rares exceptions près. La protection par droit d'auteur de la structure des programmes fait l'objet de jurisprudences contradictoires. Il faut toutefois nuancer l'efficacité de la protection par copyright. En cas d'accusation de violation de copyright, beaucoup de sociétés ripostent en invoquant l'utilisation du copyright à des fins non concurrentielles, et notamment qui contreviendrait aux lois antitrust (Robert Merges, 1996, p. 285).

Concernant les possibilités de breveter des logiciels c'est aux Etats-Unis que l'évolution a été la plus importante. Alors que les copyrights dépendent d'un organe législatif (la Librairie du Congrès), les brevets émanent d'un service gouvernemental relevant du Département du Commerce. La position de départ du *Patent and Trademark Office* (TPO) était hostile à la délivrance de brevets pour des programmes d'ordinateurs, en raison de leur nature assimilable à un enchaînement d'étapes abstraites et/ou d'algorithmes mathématiques. Jusqu'au début des années quatre-vingt la Cour Suprême, en s'en tenant au principe de lier les algorithmes informatiques à des "lois de la nature" et à des principes scientifiques, refusait les brevets sur les logiciels et allait jusqu'à invalider certains brevets. Mais le changement de la perception des problèmes des droits de propriété en général et le fait que la Cour Suprême ait cédé la juridiction à la *Court of Appeals for the Federal Circuit*, créée en 1982 et favorable aux

brevets, ont modifié la situation⁴¹ (Robert Merges, 1996, p. 280). A partir des années quatre-vingt le nombre des brevets accordés croit très fortement. Jean-Paul Smets-Solanes et Benoît Faucon estiment que le TPO accepte environ 20 000 brevets sur des procédés liés au logiciel chaque année (à titre de comparaison 70 000 brevets sont déposés en France par an tous domaines confondus). A elle seule IBM a déposé 1 087 brevets aux Etats-Unis en 1993, dont 20 % concernaient des logiciels (01 Informatique, n° 1444, 14-3-97). Ces brevets concernent des processus contrôlés par ordinateurs, des méthodes de gestion implantées sur ordinateurs, des interfaces homme-machine, et des algorithmes implémentés par ordinateur (Robert Merges, 1996, p. 279). En 1994 pour la première fois un brevet a été utilisé dans un procès : Microsoft s'est vu condamnée pour violation d'un brevet de compression de données détenu par Stac Electronics avec laquelle elle avait échoué sur les conditions d'un accord de licence. Microsoft a été contrainte à verser 120 millions de dollars de dommages et intérêts à Stac Electronics et à retirer provisoirement le produit en cause du marché.

d - Les problèmes de la brevetabilité des logiciels

La brevetabilité croissante des logiciels pose une série de problèmes. Théoriquement le brevet protège l'invention, une invention étant une solution nouvelle à un problème technique (pas nécessairement nouveau) qui doit satisfaire trois critères : nouveauté, inventivité et applicabilité industrielle (Jean-Alain Héraud, 1995, p. 92). L'invention est distincte de l'innovation et de la découverte scientifique qui n'est pas brevetable (non appropriable et diffusion libre). Les distinctions entre ces trois notions, déjà peu évidentes en général, sont particulièrement problématiques dans le cas des logiciels, où d'une part certains résultats de la recherche débouchent de façon quasi immédiate sur des produits opérationnels (les algorithmes de compression de données et de cryptage par exemple), et où l'évolution technologique résulte essentiellement d'un processus continu d'innovations incrémentales.

Mis à part les applications des découvertes de la recherche fondamentale, les véritables inventions sont souvent le fruit d'une idée nouvelle concernant un produit, que ce soit le produit lui-même (tableur, navigateur), ses interfaces (souris, interface graphique) ou des

⁴¹ Par exemple en 1982, le TPO avait rejeté une demande de brevet sur un système de reconnaissance du signal basé sur un nouvel algorithme. Cette demande sera acceptée par la Court of Appeals for the Federal Circuit en 1989. Cette dernière reconnaissait que l'invention concernait un algorithme, *a priori* non brevetable, mais que le brevet, accordé au programme résidant en ROM, n'empêchait pas l'utilisation de l'algorithme qui reste dans le domaine public (Robert Merges, 1996, p. 280).

fonctions nouvelles⁴². Dans ces situations où l'invention réside plus dans l'idée elle-même que dans sa mise en œuvre, le droit d'auteur qui protège seulement l'expression originale d'une idée n'est pas adapté. Mais dans les cas où les inventeurs ont essayé d'utiliser la protection par le brevet, celle-ci ne s'est pas révélée non plus efficace, l'exemple le plus célèbre étant la tentative de Xerox de faire breveter ses inventions concernant la souris et les interfaces graphiques : le brevet a été annulé par la justice américaine, qui a considéré qu'il s'agissait d'une idée et non d'un procédé, et a été accordé à Apple qui s'était pourtant fortement inspirée des travaux de Xerox dans ce domaine (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 173). Dans ce domaine, les progiciels apparaissent comme une bonne illustration du modèle de David Teece (1986) où l'innovateur tire moins de bénéfices de son innovation qu'un imitateur disposant d'actifs complémentaires des savoir-faire d'innovation (circuits de financement, de distribution...). Par exemple, dans les progiciels bureautiques pour micro-ordinateurs Microsoft est devenu progressivement hégémonique et les sociétés qui avaient développé le premier tableur (Visicalc), le premier traitement de texte (Wordstar) et le premier système de gestion de bases de données (dBase) ont toutes disparu.

Les brevets pourraient sembler plus adaptés aux inventions concernant des procédés. Mais dans ce cas, en contrepartie du monopole temporaire d'exploitation qu'il confère à son détenteur, le dépôt d'un brevet nécessite de livrer des informations et fournit des indications aux autres acteurs sur la "praticabilité" d'une direction de recherche, ses potentialités (Michel Callon, 1994 A, p. 8)⁴³. De plus, dans le cas des logiciels, les améliorations résultent essentiellement de l'application de principes généraux (donc non brevetables) et d'un flux continu d'améliorations incrémentales.

⁴² Elles correspondent à ce que André Barcet, Joël Bonamy et Anne Mayère (1987) appellent des innovations fonctionnelles (créer ou répondre à de nouveaux besoins) et dont ils ont montré l'importance dans le cas plus général des services intellectuels, ainsi que les difficultés d'appropriation et de protection.

⁴³ Cet aspect des brevets qui apparaissait comme un mal nécessaire à l'origine est de plus en plus considéré comme un avantage pour la diffusion de l'innovation comparativement à la pratique du secret (Dominique Foray, 1995, p. 139). Toutefois il peut être annulé par des pratiques de multiplication extrême de dépôts de brevets, ne correspondant pas à des perspectives économiquement intéressantes, mais visant à dissimuler la stratégie de recherche et à orienter les imitateurs potentiels vers de fausses pistes (Jean-Alain Héraud, 1995, p. 11).

De ce fait la plupart des brevets déposés concernent des questions relativement mineures⁴⁴, voire triviales⁴⁵ ou conventionnelles⁴⁶. Certes en théorie, à l'exigence d'originalité définie par la législation du copyright s'ajoute celle de la nouveauté dans le cadre du système de brevet. Mais le critère de nouveauté est difficile à fonder en raison de l'absence d'un inventaire de l'état de l'art et des pratiques antérieures de dépôts de brevets (Jean-Benoît Zimmermann, 1995 B, p. 193). Les délais d'application (32 mois en moyenne) sont en totale dysharmonie avec le cycle de vie très bref du produit (Dominique Foray, 1995, p. 139). Aux Etats-Unis, le dépôt des brevets sur les logiciels est devenu si important que le bureau américain des brevets est devenu incapable de procéder à des recherches d'antériorité fiables : par exemple, un brevet a été récemment accordé à Microsoft pour une technologie de lissage de caractères déjà publiée en 1976 (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 170).

La plupart des grandes sociétés de logiciels, qui disposent de moyens financiers importants, déposent de multiples brevets⁴⁷ dont la validité est douteuse. De toute façon, la plupart des plaintes pour contrefaçon se résolvent par des échanges de brevets et des accords croisés (Robert Merges, 1996, p. 286). Cette pratique de dépôts de brevets pour pouvoir

⁴⁴ On peut citer aux Etats-Unis, le brevet sur la technique *natural order recalc* qui permet de recalculer les conséquences sur l'ensemble d'une feuille de calcul d'une modification du contenu d'une cellule ; ce principe utilisé dans tous les tableurs fait l'objet par le détenteur du brevet d'un procès contre six éditeurs de logiciels (Jean-Benoît Zimmermann, 1995, p. 192). Autre exemple, le brevet n° US5860073 accordé le 12 janvier 1999 à la société Microsoft par l'office américain des brevets, sur "la feuille de style dans un document électronique" dont la contribution technique réside dans le fait que "contrairement aux autres systèmes, on peut définir le style d'une partie du document avant d'y avoir entré le texte" (Philippe Riviere, 1999, p. 23) ; de plus, Microsoft avait fait adopter auparavant comme standard par le World Wide Web Consortium (W3C) cette technologie de mise en page (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 169).

⁴⁵ Jean-Paul Smets-Solanes et Benoît Faucon (1999, p. 141) citent l'exemple de la fonction mathématique "ou exclusif" qui a été brevetée pour son application au procédé d'inversion des couleurs sur un écran (exemple du clignotement du curseur) et qui voit des sociétés attaquées pour contrefaçon. Ce procédé est tellement connu et élémentaire que personne n'avait osé déposer un brevet auparavant. Les mêmes auteurs mentionnent également le principe élémentaire consistant à vendre de la musique téléchargeable sur Internet qui fait l'objet d'un brevet aux Etats-Unis (p. 169).

⁴⁶ Un exemple extrême de demande de brevet, toutefois repoussé, est le cas de Lotus qui voulait protéger la combinaison de touches "/" et une lettre pour ouvrir un menu (par exemple "/"F" pour le menu File) (Robert Merges, 1996, p. 289).

⁴⁷ Pour la septième année consécutive, IBM est la société qui a déposé le plus de brevets (2756 en 1999), soit 900 de plus que son plus proche concurrent (Canon). Le portefeuille de brevets d'IBM est estimé à 30 milliards de dollars et génère des revenus de plus d'un milliard de dollar par an. Ces brevets concernent des composants matériels (par exemple les semi-conducteurs) et de plus en plus fréquemment des aspects liés à des logiciels. En 1999, ce sont surtout des questions liées au commerce électronique (méthode d'amélioration des transactions en ligne à base d'agents intelligents, système de commande en ligne à partir d'un catalogue électronique...) qui ont fait l'objet de dépôts de brevets de la part d'IBM (01 Informatique, 13-1-2000).

négozier des arrangements avec les concurrents en position de force se rencontre également dans l'industrie automobile (Jean-Alain Héraud, 1995, p. 114). Mais à la différence de l'industrie automobile, le secteur des progiciels se caractérise par la présence à côté de groupes puissants d'une multitude de petites sociétés. Celles-ci n'ont souvent pas les moyens de déposer de nombreux brevets⁴⁸ et de contester en justice la validité des brevets qui leur sont opposés : par exemple, une *start-up* française a dû cesser ses activités aux États-Unis après avoir été attaquée pour contrefaçon ; la société Adobe, attaquée ensuite pour le même brevet, a réussi à le faire annuler lors du procès (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 142)⁴⁹.

De fait, "un progiciel moderne peut contenir plusieurs milliers de traitements brevetables individuellement et constituant chacun un risque potentiel de contrefaçon d'un brevet en cours d'examen" (B. Kahin, 1990, p. 40). Enfin, la croissance du nombre de brevets sur des aspects mineurs, peut entraver le développement jugé très prometteur de la production de progiciels à partir de composants provenant de différentes sources, par l'incertitude sur l'étendue du droit de propriété existant sur tel ou tel composant (OTA, 1992, p. 154)⁵⁰.

Enfin, le développement de la brevetabilité du logiciel a pris à contre-pied ceux qui entendaient s'appuyer sur une utilisation particulière du droit d'auteur avec la création de licences de logiciels libres (cf. paragraphe suivant) pour limiter au maximum les possibilités d'appropriabilité du logiciel. Contre l'argumentation d'une protection forte favorisant l'effort inventif par l'internalisation de ses effets et par la nécessité d'entrer sur de nouveaux domaines pour échapper à ce qui est protégé (Robert Merges, 1996, p. 286), les promoteurs des logiciels libres argumentent qu'une appropriabilité plus faible peut entraîner un développement technologique plus rapide : disparition du coût des droits de propriété, facilité d'entrée et existence de technologies multiples et rivales (idem, p. 290). Si les incitations pour la recherche-développement sont plus faibles, ce phénomène serait plus que compensé par

⁴⁸ Selon Jean-Paul Smets-Solanes et Benoît Faucon, déposer un brevet international coûte près de 100 000 F par an et le défendre en cas de violation coûte plus de 250 000 F (1999, p. 169).

⁴⁹ On peut citer également l'exemple de Lotus qui a engagé des actions en justice sur le *look and feel* de son tableur en 1987 contre deux petits éditeurs (Paperback Software et Mosaic Software) et en 1990 contre Borland, mais qui ne s'est jamais attaquée à Microsoft ou à Computer Associates, auxquels elle pourrait faire les mêmes reproches mais qui sont beaucoup plus puissants (Frédéric Dromby, 1999, p. 285 et 578-579).

⁵⁰ Ce problème risque de se développer également pour les droits d'auteurs des produits multimédias comprenant de multiples fragments d'œuvres de nature très diverses (textuelles, graphiques, audiovisuelles...).

l'importance des *spillovers* (effets de report) qui permet à une entreprise d'accéder aux résultats de la recherche-développement de ses concurrents, ce qu'a mis en évidence le modèle de M. Spence (1984).

e - La multiplicité des statuts juridiques des logiciels

Les progiciels commerciaux ont donc des statuts juridiques variables dans le temps, dans l'espace (cf. c) et qui peuvent être différents pour les différents éléments qui les constituent (code, algorithme, *look and feel*, interface...). Il faut de plus ajouter le cas des logiciels sur-mesure (service individualisé non reproductible) auxquels, sur un plan juridique, s'appliquent classiquement les droits des contrats et le secret des affaires, et prendre en compte le phénomène particulier constitué par les logiciels libres, que l'on peut analyser comme une volonté de maintenir les logiciels comme biens collectifs. Les racines de ce phénomène remontent aux premiers temps de l'informatique avec, dans la communauté universitaire, la tradition d'échange bénévole de programmes considérés comme une création de l'esprit qui doit échapper aux circuits marchands (Gérard Dréan, 1996 A, p. 206). Cette tradition s'est perpétrée sous deux aspects différents.

D'une part, au début de la microinformatique, la pratique de copies libres de logiciels au sein de clubs d'utilisateurs, à laquelle s'opposent violemment les éditeurs de progiciels commerciaux pour micro-ordinateurs (notamment Bill Gates en 1976), donnera naissance à la production de logiciels en *freeware* (ou "gratuciels") qui sont fournis gratuitement aux utilisateurs, et de logiciels en *shareware* (ou "distribuciels") qui sont diffusés librement, l'utilisateur intéressé par le produit étant tenu moralement de verser une contribution, le plus souvent modeste, à l'auteur du logiciel.

D'autre part, se développe la création de logiciels libres, dont la particularité fondamentale est que leur code source est librement accessible, redistribuable et modifiable. En effet, dans le cas des progiciels commerciaux, comme le plus souvent des logiciels en *shareware* et en *freeware*, c'est uniquement le code objet du logiciel qui est fourni à l'utilisateur, sans possibilité de connaître le code source du logiciel. Sur le plan juridique les logiciels libres sont basés sur des licences particulières, dont la plus utilisée est la Licence Publique Générale (GPL). Cette licence est l'œuvre de la *Free Software Foundation* (FSF), dont l'objectif est de soutenir le développement de logiciels libres. Cette fondation créée en 1985 par Richard Stallman (un informaticien du MIT) se situait dans le prolongement du

projet GNU⁵¹ lancé l'année précédente dont l'objectif était de créer un Unix libre de droit face aux différentes versions commerciales d'Unix des différents constructeurs informatiques. Ces licences publiques utilisent à rebours le droit d'auteur⁵² pour garantir qu'un logiciel libre le restera : toute personne peut utiliser, distribuer, modifier librement un logiciel sous licence GPL à condition de maintenir libre ce logiciel, c'est à dire de fournir librement le code source⁵³. C'est la différence avec un logiciel du domaine public qui n'étant pas couvert par le droit d'auteur peut faire l'objet de dépôt de licence classique⁵⁴. Les licences publiques posent toutefois un problème par rapport au droit européen. En effet, les licences publiques, comme la plupart des contrats de licence pour des logiciels aux Etats-Unis, ne comportent aucune garantie et lèvent toute responsabilité de la part de l'éditeur. Ceci est en contradiction avec une directive européenne du 25 juillet 1985 qui instaure une protection contre les logiciels défectueux, complétée par une directive du 5 avril 1993 concernant les clauses abusives dans les contrats conclus avec les consommateurs. Ces directives ne s'appliquent qu'à la vente de logiciels à des particuliers ou à des professionnels non informaticiens. Elles ont été confirmées en France par le Garde des Sceaux qui a précisé en août 1998 que "la responsabilité du fait des produits défectueux a vocation à englober la catégorie juridique des meubles à laquelle appartiennent les logiciels" (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 75).

Enfin, par rapport au succès relatif des logiciels libres (cf. chapitre VIII) et à la tendance à s'appuyer sur les brevets plutôt que sur les droits d'auteur pour garantir les droits de propriété, se sont développées récemment de nouvelles formes juridiques basées sur des licences spécifiques (*Netscape Public Licence, Sun Community Source License, Jini Technology Public Licence...*) combinant publicité du code source des logiciels, versement de

⁵¹ GNU est un acronyme récursif signifiant GNU's Not Unix.

⁵² Richard Stallman parle de *copyleft* !

⁵³ Toutefois à côté de la licence GPL, il existe une licence plus souple (Library GPL) destinée aux bibliothèques de composants logiciels : celle-ci autorise l'utilisation de composants logiciels libres dans des logiciels "privés" sans être obligé de diffuser le code source pour l'ensemble du logiciel intégrant certains de ces composants. La création de cette licence, qui apparaît contradictoire avec l'éthique du logiciel libre, se justifie pour des raisons stratégiques, à savoir faciliter l'utilisation et la diffusion des composants logiciels libres (Richard Stallman, 1999).

⁵⁴ Richard Stallman (1999) cite comme exemple de privatisation d'un logiciel du domaine public, l'interface Unix, X-Windows, initialement distribuée par le MIT avec une licence très permissive et qui a été appropriée par des promoteurs des environnements Unix "propriétaires" : ceux-ci ont intégré cette interface à leur système sous forme de code binaire exécutable sans diffuser le code-source.

redevances en cas d'utilisation commerciale, et procédures de certification. Ces licences s'accompagnent de dépôts de brevets effectués par ces sociétés⁵⁵.

En conclusion, le caractère de bien partiellement collectif des progiciels a débouché sur des statuts juridiques variés et non stabilisés, les catégories juridiques habituelles ayant du mal à s'adapter aux spécificités des logiciels. Nous allons rencontrer d'autres difficultés pour appliquer les théories économiques destinées à expliquer la valeur et les prix des produits.

C - LES DIFFICULTES POUR APPREHENDER LA VALEUR DES LOGICIELS

La détermination des prix des logiciels apparaît à bien des égards singulière : très grande variabilité des prix dans le temps et selon les utilisateurs pour les progiciels, domination de la "régie" et difficultés à instaurer une tarification au "forfait" pour les logiciels sur mesure (1). Pour les logiciels, les différentes théories de la valeur élaborées pour expliquer les prix des marchandises, qu'elles soient fondées sur la valeur d'usage du produit (2) ou sur la quantité de travail (3) apparaissent inadaptées.

1 - Les singularités de la détermination des prix des logiciels

Une particularité des logiciels est que le système des prix apparaît comme un argument relativement secondaire de la prise de décision (importance de la compétitivité hors prix) , comme un vecteur d'information assez pauvre, à l'encontre du postulat hayekien, que ce soit dans le cas des progiciels (extrême variabilité) ou des logiciels sur mesure (tarification en "régie").

a - L'extrême variabilité des prix des progiciels

Celle-ci s'explique fondamentalement par la nature de bien intangible, et les caractéristiques de biens collectifs qui en découlent, des progiciels. Sans reprendre l'analyse effectuée précédemment, il faut souligner l'importance des coûts irrécupérables (l'essentiel des dépenses concerne le développement de l'original, sont donc engagées avant de vendre le premier exemplaire et ne peuvent être récupérées en cas de renoncement au projet, comme pour les biens matériels où il est en général possible de les revendre) et les difficultés

⁵⁵ Ces formes de logiciels "pseudo-libres" ont suscité la création de l'*Open Source Initiative*, présidée par Eric S. Raymond, dont l'objectif est de défendre le label "logiciel libre" contre les tentatives de récupération (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 127).

d'appropriation. Ceci explique qu'il soit fréquent d'inclure la vente du progiciel soit dans la vente du matériel⁵⁶, soit dans la vente d'un autre progiciel plus indispensable⁵⁷, soit (plus récemment) dans la vente de services, sans que le prix du progiciel considéré apparaisse clairement pour l'acheteur. Il faut également remarquer qu'avec le développement de la vente de progiciels par Internet, le prix réel payé par l'utilisateur n'est connu qu'après coup, dépend des compétences et de l'équipement de l'acheteur, et est une combinaison, souvent indistincte, de redevances, de charges d'utilisation et de coûts de communication (Herbert R. Brinberg, Thomas E. Pinelli et Rebecca O. Barclay, 1997, p. 184).

L'importance des coûts fixes, indépendants des quantités vendues, explique que le prix d'un produit, élevé lors de son lancement pourra baisser fortement avec l'élargissement de sa diffusion, l'élasticité-prix de la demande étant élevée (OCDE, 1997 B, p. 22) : par exemple, Eurostaf estime que le prix d'un *firewall* (logiciel de protection d'un réseau interne connecté à l'extérieur) devrait baisser de 16 000 dollars à 650 dollars entre 1996 et 2000, ses ventes prévues passant de 40 000 unités à 1510 000 sur la même période (1997 A, p. 25). Il faut également prendre en compte les stratégies des producteurs liées à la force des rendements croissants d'adoption dans la diffusion des progiciels. Pour s'imposer sur un marché, une entreprise peut commercialiser un progiciel à un prix extrêmement bas, voire le distribuer gratuitement (stratégie de Microsoft pour imposer son navigateur sur un marché dominé à l'époque par Netscape), en espérant compenser le manque à gagner par une augmentation des prix quand il se sera retrouvé en position hégémonique⁵⁸. La spécificité de ces marchés, de type "*Winner Take All*", peut susciter le développement de pratiques surprenantes et parfois contestables⁵⁹ : par exemple Microsoft facturait la vente de son système d'exploitation, de

⁵⁶ Il est par exemple extrêmement difficile d'acheter un micro-ordinateur sans un certain nombre de progiciels, même si l'acheteur les possède déjà ou ne souhaite pas les acquérir.

⁵⁷ Le cas le plus célèbre et le plus contesté est l'intégration par Microsoft de son navigateur (Internet Explorer) dans son système d'exploitation (Windows 98).

⁵⁸ Par contre, une politique de prix très bas une fois que les anticipations ont convergé sur un produit concurrent sont rarement efficaces, une trop forte baisse de prix pouvant même être interprétée comme le signe que l'entreprise est vaincue. Carl Shapiro et Hal R. Varian citent les exemples de Computer Associates avec le logiciel Simply Money distribué gratuitement (à l'exception des frais de transport) qui n'a pas soutenu la concurrence de Quicken et de Money, d'OS/2 bradé à 50 dollars par IBM sans succès face à Microsoft, ou du tableur de Borland face à Lotus et Microsoft (1999, p. 252).

⁵⁹ "Le coût élevé de production du premier exemplaire d'un bien ou d'une technologie d'information engendre inévitablement des stratégies de différenciation du prix et du produit. Les stratégies de personnalisation à grande échelle, de discrimination par le prix, de personnalisation du contenu et du produit sont le résultat

1990 à 1994, aux constructeurs de micro-ordinateurs en fonction du nombre total de micro-ordinateurs vendus, indépendamment du fait que le micro-ordinateur vendu comprenne le système d'exploitation de Microsoft, un système d'exploitation produit par une autre société ou aucun système d'exploitation⁶⁰. De même, la rapidité avec laquelle apparaissent les nouveaux produits et les nouvelles versions, et l'insignifiance du coût marginal d'un nouvel exemplaire, peuvent conduire les entreprises à "brader" littéralement certains produits : il n'est pas rare de voir certains produits, vendus plusieurs milliers de francs, être proposés par certains canaux de distribution quelques dizaines de francs quelques années après leur lancement⁶¹.

La deuxième dimension de la variabilité des prix des progiciels réside dans la pratique d'une tarification différenciée selon le type d'utilisateurs⁶². Celle-ci peut prendre des formes multiples : rabais consentis aux possesseurs d'une ancienne version du même produit ou d'un produit concurrent, prix beaucoup plus faibles lors d'achat en grande quantité⁶³, prix différents selon le type de matériel auquel est destiné le progiciel⁶⁴ ou selon le type d'utilisation présumée, fourniture de prestations complémentaires "gratuites" d'importance différente selon

naturel de cette structure de marché. Ces stratégies peuvent cependant mettre la concurrence en question de façon sensible" (Carl Shapiro, Hal R. Varian, 1999, p. 263).

⁶⁰ Sommé par la justice de renoncer à imposer ce type de pratique aux constructeurs, Microsoft a adopté une double tarification qui apporte peu de changements dans les faits : un tarif peu élevé pour les constructeurs qui continuent à pratiquer l'ancien système (nombre de machines vendues) et un tarif beaucoup plus élevé pour les constructeurs qui ne paieraient que les systèmes d'exploitation Microsoft fournis aux utilisateurs. Dans une situation où la concurrence sur les prix est devenue particulièrement rude entre les constructeurs et où les marges se sont érodées, cette dernière possibilité n'a pas été adoptée par beaucoup de constructeurs. Microsoft ne semble pas vouloir abandonner ce type de pratiques, comme l'illustre les récentes propositions aux établissements scolaires d'une tarification ("licence School pour l'éducation") basée sur le nombre d'ordinateurs de chaque établissement scolaire ("ordinateurs éligibles") indépendamment des logiciels installés sur ces ordinateurs, et aux universités ("licence Campus") d'un prix basé sur le nombre d'enseignants et d'étudiants.

⁶¹ En France, une revue au titre significatif ("Presq'offert") s'est spécialisée dans ce type de ventes.

⁶² Aux Etats-Unis, la discrimination par les prix est illégale selon le *Robinson-Pattman Act* de 1936 lorsqu'elle entraîne "une réduction effective de la concurrence" ; toutefois il est possible de fixer des prix différents si les coûts de production sont différents, pour faire face à la concurrence. Ce n'est que si cette pratique aboutit à réduire la concurrence qu'elle peut engendrer des sanctions juridiques (Carl Shapiro, Hal R. Varian, 1999, p. 264).

⁶³ Le même produit (la suite bureautique Office de Microsoft) peut se voir proposé à un prix variant de 50 F. pour les "grands comptes" à 4 000 F. pour une P.M.E. (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 112).

⁶⁴ De façon générale, un progiciel destiné à être utilisé sur une station de travail ou sur un miniordinateur est vendu beaucoup plus cher qu'une version identique pour un micro-ordinateur.

les clients⁶⁵. Cette tarification par segments est permise par le fait que pour les progiciels "le contrat de licence accorde à l'utilisateur un droit d'usage définitif, sans lui en transférer la propriété dans la mesure où il ne peut le revendre à un tiers" (Gérard Dréan, 1996 A, p. 189). De ce fait, il devient possible pour le producteur de pratiquer une discrimination par les prix (Godefroy Dang Nguyen, 1995, p. 215-216), qui en ajustant plus ou moins finement les prix à la propension à payer de chaque segment de clientèle, permet de récupérer une partie du surplus des consommateurs (la différence entre le prix qu'étaient prêts à payer certains consommateurs et le prix unique du produit dans le cas d'une impossibilité de discrimination). Une autre possibilité pour arriver au même résultat consisterait à différencier le produit en l'adaptant à l'utilisation distincte des consommateurs (Jean-Pierre Angelier, 1997, p. 118), ce qu'avait fait par exemple Microsoft en développant une version aux fonctionnalités réduites de son traitement de texte (Word Junior). Elle présente toutefois l'inconvénient de développer et de gérer simultanément plusieurs produits différents et donc de ne pas bénéficier pleinement des économies d'échelle possibles⁶⁶. Dans le cas des progiciels, il est possible et souvent plus rentable de vendre le même produit à des prix différents selon des profils d'utilisateurs⁶⁷, ce qui constitue une des explications du fait que fréquemment une partie minime des fonctionnalités d'un progiciel est effectivement utilisée.

⁶⁵ Cette pratique était l'œuvre d'IBM dans les années soixante pour les ordinateurs lorsqu'elle ne facturait pas les développements logiciels. Elle disposait ainsi "d'un moyen sans faille de modulation de ses prestations "à la tête du client", car il était difficile de mesurer ce que lui coûtait réellement ce service, qu'elle accordait de façon préférentielle à ses meilleurs clients" (Godefroy Dang Nguyen, 1995, p. 329).

⁶⁶ Dans le domaine du matériel la possibilité de concilier économie d'échelle et discrimination des prix est parfois réalisée par des artifices techniques : par exemple, pour certaines gammes d'ordinateurs IBM ne produisait et ne livrait que le plus puissant qu'elle "bridait" éventuellement (pratique appelée du "tournevis d'or"). De façon plus générale, il était fréquent qu'une machine moins puissante, et donc offerte à un prix inférieur, soit obtenue en ajoutant à la machine initiale un dispositif de ralentissement ou autre limitation, ce qui augmente donc son prix de revient par rapport à la machine initiale ; par exemple, l'imprimante LaserPrinter série E d'IBM est fonctionnellement identique à la LaserPrinter standard vendue plus cher, mais comprend en plus une puce programmée pour diviser par deux le temps d'impression (Carl Shapiro, Hal R. Varian, 1999, p. 58). Gérard Dréan parle de tarification "fonctionnelle" pour désigner cette pratique, qui "permet au constructeur d'empocher la totalité de la différence entre la valeur d'usage que présente le produit pour son utilisateur et le coût de revient de ce produit" (1996, p. 18).

⁶⁷ Toutefois il existe parfois également des versions "dégradées" de logiciels où certaines fonctions ont été désactivées (ce qui signifie par ailleurs qu'elles sont plus coûteuses). Carl Shapiro et Hal R. Varian citent les exemples du logiciel Mathematica, pour lequel il existe une version "étudiant" où certaines fonctions ont été désactivées pour ralentir l'exécution et la visualisation des graphiques, le logiciel PhotoDeluxe version "allégée" de Photoshop d'Adobe, et la version Windows NT Workstation qui est la même que la version Serveur, beaucoup plus chère, mais qui est "bridée" ce que confirme, malgré les dénégations de Microsoft, le fait que certains administrateurs de réseaux aient réussi à la "débrider" (1999, p. 63 - 69).

L'ensemble de ces pratiques explique que les prix des logiciels "sont à peu près indépendants du contenu fonctionnel du produit et donc de ses coûts" (Gérard Dréan, 1996 A, p. 223). Enfin, il faut noter que se développe la location des logiciels en ligne effectuée par les *Application Service Providers* (ASP), par exemple la société Hot Office Technologies qui loue un logiciel semblable à Lotus Note (Les News, 13/09/99). Des acteurs puissants (Sun, Microsoft) ont annoncé leur intention de recourir à ce mode de commercialisation. Le succès de ce modèle économique, où le logiciel ne fonctionne plus sur la machine de l'utilisateur mais sur un serveur Web, est toutefois subordonné à l'avenir de la conception de l'informatique dite "*Network Centric Computing*" qui n'est nullement assuré (cf. chapitre II).

b - Les pratiques dominantes de la "régie" et les difficultés du "forfait" pour les logiciels sur mesure

La production de logiciels sur mesure est fondamentalement une activité de service. Pour les logiciels développés en interne, l'existence éventuelle d'un prix est de nature essentiellement conventionnelle. Dans les cas où ces logiciels sont développés par un prestataire extérieur, il existe deux grands modes de facturations : la pratique de la "régie" qui consiste pour le prestataire à facturer les moyens essentiellement humains mis à la disposition du client qui demeure maître d'œuvre de la prestation (le prestataire a seulement une obligation de moyens), et la pratique du "forfait" où le prix d'une prestation précisément définie est fixé préalablement à l'exécution de celle-ci (obligation de résultats : le prestataire doit réaliser un projet donné en respectant les coûts et les délais établis avec le client, voire même des engagements de fonctionnement et de performances). En théorie, le forfait (connaissance des caractéristiques du produit et de son prix *ex ante*) est une procédure qui correspond au rôle attribué au mécanisme des prix dans une économie concurrentielle. Pourtant dans la pratique, c'est le mode de la régie qui est largement dominant.

En effet, le développement d'un logiciel sur mesure est "un éventail étendu d'activités qui part de la naissance d'un besoin d'un utilisateur, très flou, immatériel et instable, pour se terminer par l'exploitation d'un « produit » à caractère industriel qui doit être performant, documenté et robuste à l'exploitation". Entre ces deux extrêmes, "tout un cheminement intellectuel et industriel, la démarche d'ingénierie, permet de progresser par itérations vers des représentations de l'objet « produit-service » de plus en plus matérielles, adaptées et stables." (Serge Bouchy, 1994, p. 174). Il est quasiment impossible même en utilisant des méthodes très formalisées et en développant des cahiers des charges volumineux d'établir préalablement à la prestation des spécificités exhaustives et intangibles de celle-ci. L'établissement

progressif de ces spécificités représente une part de plus en plus importante de l'activité, et les méthodes itératives plus récentes visant à mieux répondre aux besoins des utilisateurs intègrent une évolution inévitable des caractéristiques de l'application dans un processus d'interaction permanente avec les utilisateurs. Le développement de logiciel sur mesure est typiquement une activité complexe correspondant à un mandat confus (Jacques Girin, 1994, p. 19), où un contrat est nécessairement incomplet, les co-contractants n'étant pas en mesure de préciser avec exactitude, le résultat attendu, l'objet sur lequel portera l'évaluation du résultat et les moyens de vérifier les engagements, d'autant plus que le mandant participe à la prestation et a donc une part de responsabilité dans la réalisation de celle-ci.

De ce fait, il est assez inévitable que les contrats soient centrés sur la définition des moyens mis à disposition (modèle de la régie), rares étant les entreprises qui travaillent réellement sur la base de contrats de résultats forfaitaires (Gérard Dréan, 1996 A, p. 279). L'existence de prestations au forfait est souvent de la "régie déguisée" avec une succession de forfaits qui évoluent au fur et à mesure du développement de la prestation. C'est notamment le cas pour les projets de taille importante où certes existent des études de faisabilité et des études préalables avant la signature du contrat, mais qui concernent justement les types de projets qui voient les dérives les plus importantes par rapport aux spécifications initiales⁶⁸.

Pour l'analyse des contrats en matière de développement de logiciel sur mesure, il faut prendre en compte l'importance de l'asymétrie informationnelle et les problèmes d'incertitude (S. Wang, 1992), les phénomènes d'*aléa moral* et de *sélection adverse*, prenant une dimension majeure. Dès lors le prix ne peut être un indicateur suffisant pour orienter le comportement des clients dans la concurrence⁶⁹. Pour que les prix déterminent l'attitude des consommateurs, il est nécessaire qu'il y ait homogénéité du produit. Or, même lorsqu'il existe plusieurs propositions substituables au sens où elles visent à répondre aux mêmes besoins, elles ne sont pas homogènes car elles ne présentent pas en général les mêmes caractéristiques. La

⁶⁸ Un exemple des difficultés de tels contrats est la rupture spectaculaire survenue en pleine réalisation d'un projet entre la Bibliothèque Nationale de France et Cap Gemini sur un contrat d'un montant initial de 138 millions de francs. La BNF reproche le manque de fiabilité et les retards de livraison des premières versions (jusqu'à 22 mois), Cap Gemini mettant en avant les modifications du cahier des charges (accroissement des fonctionnalités qui impliquerait une surcharge de travail de 50 %) (01 Informatique, 30/07/1999).

⁶⁹ Pour les services de façon plus générale, Jacques De Bandt et Jean Gadrey s'interrogent : "dans quelles conditions peut-on encore parler de marché (compte tenu de l'incertitude sur l'objet et l'enjeu de la transaction), de prix (vu la difficulté à définir des unités de produits), de concurrence (lorsque les performances sont évaluées de façon anticipée) ?" (1994, p. 16)

production des logiciels sur mesure combine les caractéristiques des différents cas où, selon Jacques Girin, la question de la connaissance de la compétence du fournisseur ne peut être éludée : mandat nouveau à une entité sans possibilité d'attendre les résultats pour s'assurer des compétences du mandataire, caractère déraisonnable du fait d'attendre les résultats pour évaluer la capacité d'un système à les atteindre, conception des traits généraux du "design" d'un sous-ensemble en vue d'une tâche déterminée, question de la performance qui ne se pose pas de manière simple et qui ne peut se résumer facilement dans quelques indicateurs physiques ou financiers (1994, p. 26). La compétence du prestataire peut être particulièrement difficile à évaluer directement par le client et des phénomènes de nature conventionnelle (réputation, confiance, certification) vont modeler fortement les comportements dans le secteur.

2 - Les difficultés pour appréhender la valeur d'usage d'un logiciel

La notion de valeur d'usage est au centre de l'explication des prix par la théorie néoclassique de base avec les notions de coût marginal croissant et de rareté. La question de la rareté se pose de façon particulière pour les logiciels, comme de façon plus générale pour les biens intangibles existant sous forme numérique (Blaise Cronin, 1997, p. 15-16), dans la mesure où la rareté ne peut être créée qu'artificiellement, en conséquence d'un coût marginal qui tend vers zéro avec les quantités produites. "La production d'un bien d'information nécessite des coûts fixes élevés, mais a un coût marginal faible", ce qui implique "qu'il est impossible de fixer le prix d'un tel produit en se basant sur le coût de production" et "qu'il faut fixer le prix en fonction de la valeur du bien pour les consommateurs" (Carl Shapiro, Hal R. Varian, 1999, p. 9-10)⁷⁰. La question que nous voudrions aborder maintenant est celle de la difficulté qu'il y a à appréhender la valeur d'usage d'un logiciel⁷¹, indépendamment du caractère pertinent de ce cadre théorique dans l'explication de la détermination des prix pour l'ensemble des biens et services.

⁷⁰ C'est selon ces auteurs ce qui explique que "le prix sera nécessairement fixé de façon discriminante", puisque "les individus valorisent très différemment les biens d'information" (Carl Shapiro, Hal R. Varian, 1999, p. 10).

⁷¹ Par certains aspects, le logiciel est confronté à des difficultés identiques que le produit de la recherche-développement qu'Olivier Weinstein a analysé en soulignant "l'indétermination de sa valeur d'usage" (1989, p. 82).

a - Les difficultés à isoler l'impact d'un composant d'un bien-système mobilisant les compétences des utilisateurs

La première difficulté est qu'un logiciel est nécessairement utilisé de façon indissociable d'autres produits logiciels et matériels qui déterminent les caractéristiques de son environnement (cf. chapitre I). Cette situation est poussée à l'extrême dans le cas des "systèmes propriétaires" où un logiciel ne peut fonctionner que dans un type d'environnement matériel et logiciel limité. L'ouverture des systèmes a atténué ces dépendances, mais il n'en demeure pas moins que ce que peut appréhender l'utilisateur c'est l'efficacité globale d'un système constitué de composants matériels, logiciels et de services. L'évolution des logiciels, qui de plus en plus réalisent des fonctions complexes, renforce cette tendance. Ce qui devient déterminant c'est plus l'efficacité de l'interopération entre les différents composants que l'efficacité de chaque composant pris isolément. Par exemple, un échec dans une recherche d'informations sur le Web peut résulter d'une défaillance de la coordination des différents éléments concernés très divers⁷² ou d'un des éléments du système, sans possibilité pour l'utilisateur d'identifier le produit incriminé.

De plus la réalisation des performances potentielles du système dépend de manière importante des compétences de l'utilisateur et de la qualité des services qui contribuent à développer ces compétences. Selon Paricia D. Fletcher et Lester Diamond (1997, p. 176), "la véritable ressource réside dans les hommes qui sont capables d'exploiter les technologies de l'information". A la notion de valeur d'usage incorporée dans le produit lors de sa création, d'une utilité objectivée intrinsèque au produit, Orio Giarini et Walter R. Stahel (1990) proposent de substituer la notion de valeur d'utilisation d'un système, qui est l'utilité obtenue d'un système pendant l'ensemble de sa durée de vie, intégrant le rôle actif du consommateur, caractérisé comme étant un *prosumer* (consommateur-producteur) par Alvin Toffler. Cette valeur dépend des performances d'un agencement organisationnel particulier, composite constitué d'éléments hétérogènes (matériels, logiciels, humains) reliés entre eux (Jacques Girin, 1994, p. 25). Elle est difficilement mesurable objectivement et ne peut le plus souvent qu'être évaluée par des "mesures d'ancrage" ou indicateurs opérationnels de performance qui

⁷² Sans prétention à l'exhaustivité, on peut mentionner le matériel et les logiciels (système d'exploitation, outils de navigation et de communication) du "client", du "serveur", et les différents composants matériels et logiciels qui constituent l'infrastructure du réseau.

dépendent des objectifs de l'organisation (Herbert R. Brinberg, Thomas E. Pinelli et Rebecca O. Barclay, 1997, p. 188).

b - L'insuffisance de l'appréciation par les caractéristiques d'usage directes et la nécessité de recourir aux caractéristiques d'usage indirectes

De ce point de vue les indicateurs pertinents pour l'organisation concernent moins les caractéristiques d'usage directes (ou efficacité) que les caractéristiques d'usage indirectes (ou effectivité).

Les caractéristiques d'usage directes sont tout d'abord difficiles à mesurer : à la différence des composants matériels où l'on peut effectuer des mesures physiques des différentes dimensions de l'efficacité, les techniques de mesure (les "métriques") des facteurs de qualité, exemptes autant que faire se peut de subjectivité, ne progressent que très lentement pour les logiciels (Serge Bouchy, 1994, p. 193). Ensuite, ces caractéristiques évoluent dans le temps en fonction de plusieurs facteurs non indépendants : les logiciels sont des "produits évolutifs, pour lesquels l'échange perdure au-delà de la première acquisition pour intégrer les possibilités de leur évolution" (Anne Mayère, 1997 A, p. 131), les environnements matériels et logiciels se modifient, les compétences des utilisateurs changent. Par exemple, Francis Pavé (1989) à partir d'une étude sociologique d'une dizaine de projets informatiques constate que les services utilisateurs rejettent, neutralisent ou dévoient le système, et conclut que les projets réussis sont les projets qui ont été dévoyés car ils ont été réappropriés par les agents. D'autres facteurs interviennent de façon plus indirecte : l'utilité d'un logiciel pour son détenteur peut être modifiée par son adoption par d'autres acteurs, de façon positive (rendements croissant d'adoption et externalités de réseaux) ou négative quand sa possession exclusive était un facteur de compétitivité par rapport aux entreprises concurrentes (exemple des systèmes de réservation aérienne). En reprenant les distinctions établies par P. Nelson (1970), on peut dire que les logiciels s'apparentent davantage aux "*experience goods*" dont les qualités ne se révèlent qu'au cours de leur utilisation, voire aux "*credence goods*" dont les qualités sont difficilement évaluables par l'utilisateur, qu'aux "*search goods*". Dans le cas plus général des biens d'information, qui sont des biens d'expérience, "les consommateurs se fondent souvent sur la *marque* et la *réputation* du produit" (Carl Shapiro, Hal R. Varian, 1999, p. 11).

Mais, outre leurs difficultés d'évaluation, les caractéristiques d'usage directes ne sont pas l'unique déterminant des caractéristiques d'usage indirectes qui sont *in fine* celles qui sont décisives pour l'utilisateur (le logiciel se révélant fondamentalement plus ou moins utile selon

son impact, la réponse qu'il apporte aux objectifs de son acquisition). Interviennent également et de façon encore plus importante que pour les caractéristiques d'usage directes, les caractéristiques de l'environnement non exclusivement informatique et les compétences des utilisateurs pour exploiter les potentialités du logiciel. Par exemple, et même si le manque de compétences des utilisateurs n'est pas la seule cause, Eurostaf estime que les utilisateurs ne se servent en moyenne que de 20 % des fonctionnalités d'un progiciel (1996 C, p.79). Les trois dimensions principales des caractéristiques d'usage indirectes sont l'augmentation de la productivité consécutive à l'introduction du logiciel (ou productivité indirecte, Jacques De Bandt, 1995, p. 162), l'amélioration de la qualité des actions des utilisateurs, le raccourcissement des délais de production et de livraison des utilisateurs⁷³. Ce qui va rendre particulièrement problématique la connaissance de la valeur d'usage d'un logiciel par l'utilisateur, c'est que les caractéristiques d'usage indirectes sont encore plus délicates à évaluer, ce qu'illustre les discussions sur le paradoxe de Solow à propos des effets de l'informatique.

c - Les effets de l'informatique et le "paradoxe de Solow"

L'analyse s'est tout d'abord focalisée sur les caractéristiques d'usage concernant la productivité indirecte : "l'enjeu du logiciel est justement la rationalisation qu'il permet de l'activité économique et notamment du tissu industriel. Concevoir un logiciel, c'est inventer un procédé destiné à améliorer la productivité d'une fonction donnée" (Christine Babelon, 1987, p. 7). Mais le fait qu'une part importante des logiciels (logiciels systèmes, outils de développement, bases de données communes aux applications, bibliothèques de composants...) n'ont de valeur que par rapport aux logiciels applicatifs qu'ils permettent de produire et de faire fonctionner, et la nécessité de prendre en compte également l'infrastructure matérielle nécessaire (cf. a), font que les analyses ont porté essentiellement sur l'impact en termes de productivité de l'informatique prise globalement.

Dans les années soixante-dix et quatre-vingt, la plupart des études prévoyaient que l'utilisation de l'informatique allait entraîner une forte augmentation de la productivité, notamment dans le travail de bureau, un travail que l'ordinateur devrait transformer comme la

⁷³ En se limitant aux cas, de loin les plus fréquents, où l'action peut être effectuée de façon non informatisée. Dans les situations où cette possibilité n'existe pas (par exemple, poursuite d'une cible à partir d'entrées radar, analyse de clés cryptographiques, tomographie médicale...) l'évaluation de l'apport de l'utilisation d'un logiciel devient particulièrement problématique.

machine à vapeur et le moteur électrique avaient modifié le travail en atelier (Jean-Louis Peaucelle, 1998, p. 1). Les différentes études divergeaient sur les conséquences positives (augmentation du temps libre) ou négatives (crainte du chômage) de cette évolution de la productivité du travail, mais sa croissance semblait inéluctable. Ces gains de productivité devaient résulter de la réutilisation des données dans un grand nombre d'usages, des possibilités de leur communication à un grand nombre d'utilisateurs, de la capacité à retrouver l'information pertinente dans une grande masse de données, de l'automatisation des calculs et d'une réorganisation des processus de travail ou *Business Process Reengineering* (Jean-Louis Peaucelle, 1998, p. 2).

Toutefois, au fur et à mesure de la diffusion massive de l'informatique, des doutes vont s'exprimer sur la réalité de cette relation, doutes que résume le "paradoxe de Solow" : "*on voit des ordinateurs partout sauf dans les statistiques de productivité*". Ces doutes s'appuient sur un certain nombre d'études empiriques qui mettent en évidence une absence de concomitance entre dépense informatique et productivité à différents niveaux. Sur un plan sectoriel, on peut citer aux Etats-Unis les études de R.H. Franke (1987) pour les banques et les assurances, de L. Thurow (1987) sur les services, de S. S. Roach (1989) sur les cols blancs. Pour la France, une évaluation de seize applications dans les principaux ministères, effectuée à la demande du Comité Interministériel de l'Evaluation des Politiques Publiques (Commissariat Général du Plan, 1992) souligne la faiblesse des gains de productivité réalisés. Le constat qu'entre 1982 et 1990 ce sont les emplois de secrétaires qui ont le plus augmenté en France en volume (+ 256 000, soit + 35 %, source INSEE) malgré l'introduction massive de la bureautique va dans le même sens. Les observations macroéconomiques effectuées dans différents pays montrent également une absence de lien entre une augmentation faible et erratique de la productivité apparente du travail et une croissance forte et continue de la dépense informatique en valeur, sa croissance en volume étant encore plus rapide en raison de l'importance de la baisse des prix (cf. pour la France, sur la période 1986-1995, Jean-Louis Peaucelle, 1997, p. 16). Pour l'ensemble des pays de l'OCDE, le PIB par tête qui croissait au rythme de 3,8 % l'an entre 1960 et 1973, n'augmentait plus que de 1,4 % l'an entre 1979 et 1993 ; de plus ce phénomène était plus marqué pour les services (de 2,7 % à 0,8 %) que pour l'industrie (de 5 % à 2,9 %) alors que ce sont les services qui ont connu l'informatisation la plus importante (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 65).

Au niveau microéconomique, Jean-Louis Peaucelle à partir de l'examen des différentes études menées qui présentent des résultats contradictoires, d'amplitude limitée et difficilement

interprétables, conclut que "l'impact économique de l'informatique doit être faible et, avec nos méthodes actuelles, on ne le distingue pas nettement" (1997, p. 18). Ce paradoxe n'est pas seulement un phénomène statistique mais recouvre également les interrogations des utilisateurs, organisations ou individus, sur l'efficacité réelle apportée par les technologies de l'information et de la communication (Alain Rallet, 1997, p. 85).

Les explications du paradoxe de Solow

Les explications de ce paradoxe apparent ont suscité de nombreux débats. Une interrogation préalable porte la réalité statistique du paradoxe en émettant des sérieux doutes sur la pertinence des statistiques de productivité, notamment dans les activités de services (Jean Gadrey, 1996 A), où "les mesures existantes de productivité [sont] inadaptées pour saisir la forte complexification" des activités (idem, p. 227) ou donne des résultats contradictoires⁷⁴. De même, Pascal Petit estime que "après deux décennies de débat sur le ralentissement des gains de productivité, la qualité de la mesure des gains de productivité reste une des explications principales mises en avant" (1998, p. 347). Au-delà, plusieurs explications, qui ne sont pas nécessairement contradictoires, ont été proposées.

Une première série d'explications intègre l'existence de gains de productivité qui semblent considérables pour des activités isolées et bien identifiées. Parmi de multiples exemples, on peut citer l'informatisation de la demande des cartes d'identité, passeports et cartes grises de la préfecture de Seine et Marne qui a fait passer le nombre de dossiers traités par jour par chaque employé de 25 à 150 (Le Monde Informatique, 8 avril 1994). De façon plus générale, Jean Gadrey souligne que "de très importants gains de productivité ont été (et seront encore) réalisés, dans des conditions qui peuvent parfois donner lieu à des mesures précises, dans le vaste ensemble des tâches consacrées au traitement d'informations codées, pour un volume donné d'informations de complexité donnée, à traiter sur un mode standardisé : transactions et dossiers standards, production de comptes et de ratios, opérations statistiques, mailings, facturations, etc." (1996 A, p. 213). De multiples raisons peuvent expliquer que ces gains partiels réels ne soient pas plus répandus et semblent s'évanouir quand la productivité est appréhendée à un niveau plus global : le fait que souvent la solution informatique s'ajoute

⁷⁴ Par exemple pour l'activité bancaire, pendant une période de forte informatisation, la productivité en valeur (mesurée à partir du produit bancaire) reste relativement stable semblant confirmer un impact limité de l'informatisation, alors que la productivité en nature (mesurée à partir du nombre de comptes) augmente très fortement (Jean Gadrey, 1996 A, p. 111).

à la solution non informatique au lieu de la remplacer⁷⁵, l'élasticité des tâches (Jean-Louis Peaucelle, 1998, p. 6) qui fait que des tâches obligatoires mais non effectuées auparavant faute de temps peuvent être réalisées, la nécessité de nouveaux emplois et de nouvelles tâches liées à l'informatisation⁷⁶, l'amélioration de la productivité du travail pour des postes qui ne constituaient pas des goulets d'étranglement (Jean-Louis Peaucelle, 1998, p. 7) et donc sans incidence globale en l'absence d'une modification de l'organisation du travail, le fait, lié au précédent, que l'effectif de certains postes ne peut descendre en dessous de l'unité et qu'il est déterminé en fonction des périodes de pointe.

La deuxième série d'explications est de nature plus sociologique et insiste sur la persistance de coûts liés aux limites cognitives des individus et des collectifs et aux comportements stratégiques des acteurs qui peuvent avoir intérêt à limiter l'accès à certaines informations (Eric Brousseau, 1997, p. 49). L'informatisation initiale, réalisée en renforçant la centralisation de l'information, dépossédait les utilisateurs de la maîtrise de ressources informationnelles. L'irruption d'une micro-informatique décentralisée a permis une réappropriation d'une partie de l'information, mais en se réalisant de façon quelque peu anarchique, elle a multiplié les saisies et les traitements individuels, source de coûts additionnels (Marie-Christine Monnoyer-Longe, 1997, p. 111). Symétriquement, des possibilités apportées par la mise en réseau d'équipements informatiques pour une communication étendue d'informations détenues centralement, ne sont pas exploitées, par crainte de l'autonomisation des salariés qui pourrait en résulter. Des coûts supplémentaires peuvent également provenir d'une tendance exagérée à privilégier la forme sur le contenu, et sont difficilement contrôlables en raison de l'indétermination des usages possibles de l'information et de la déstabilisation constante des procédures d'évaluation et de contrôle par l'innovation technologique (Alain Rallet, 1997, p. 95). Il en résulte une "imprévisibilité fondamentale des effets de productivité en raison de la nature intrinsèquement sociale du processus d'informatisation" (François Pichault, 1990, p. 170).

La troisième série d'explication met en cause la mauvaise gestion de l'informatisation (Erik Brynjolfsson, 1993), notamment des erreurs de management qui conduisent à continuer

⁷⁵ Thomas K. Landauer cite comme exemple le courrier électronique qui ne se substitue pas au téléphone et au courrier classique (1997, p. 5).

⁷⁶ Le Gartner Group estime que les utilisateurs consacrent une heure par jour à des tâches liées à l'outil informatique et non au contenu de leur travail.

des projets non performants mais dont l'impact est difficile à mesurer pour les gestionnaires. La modestie des gains de productivité serait la conséquence d'un gaspillage de ressources et des dysfonctionnements (Pierre Bonnaure, 1996, p. 67). Dans certaines entreprises et branches de services, il existerait un suréquipement en ordinateurs et en systèmes d'informations et une surcharge informationnelle (Jean Gadrey, 1996 A, p. 151), qui pourrait n'être que transitoire et s'expliquer par la nécessité d'une période d'apprentissage avant d'utiliser valablement les potentialités de technologies nouvelles.

La quatrième série d'explications considère qu'en fait l'objectif d'informatisation d'une organisation n'est pas la recherche de gains de productivité (et donc qu'il n'y aurait pas de paradoxe). Au niveau des administrations, la productivité (mesurée par personne) n'augmenterait que faiblement, car les effectifs seraient plus déterminés par l'histoire et la volonté des supérieurs hiérarchiques de maintenir le nombre de leurs collaborateurs, indication de l'étendue de leur pouvoir, que par la durée des tâches à accomplir (Jean-Louis Peaucelle, 1998, p. 6). Au niveau des entreprises l'informatisation peut être un investissement obligatoire en raison de nécessités administratives (initiatives des pouvoirs publics), des exigences des clients ou des entreprises partenaires (notamment concernant les modalités de transmission d'informations), et des pressions des entreprises concurrentes⁷⁷. Dans ce cas, l'informatisation améliorerait la compétitivité, en général très provisoirement vu les possibilités d'imitation, pour l'entreprise concernée mais n'aurait pas d'effet au niveau de la branche ou de l'ensemble de l'économie (Alain Rallet, 1997, p. 93), les gains en termes de part de marché s'effectuant dans un jeu à somme nulle sur le plan de la productivité, comme pour la publicité (Eric Brousseau, 1997, p. 62). Beaucoup d'innovations "centrées aujourd'hui sur des logiques de différenciation de produit pouvant aller jusqu'à un degré extrême de superficialité, l'introduction plus rapide des nouveautés, l'anticipation et le contournement des dispositifs réglementaires" présenteraient "des taux élevés de rendements privés" mais ne produiraient que "fort peu d'externalités positives" (Dominique Foray, Bengt Ake Lundvall, 1997, p. 24). Il faut toutefois noter que les études empiriques de Claude Salzman (1982) sur 150 entreprises françaises et de Paul A. Strassmann (1990 et 1997) sur des entreprises américaines ne confirment pas l'influence de l'importance des dépenses informatiques sur la rentabilité des entreprises, qui devrait résulter d'une compétitivité accrue.

⁷⁷ Il s'agit alors moins d'évaluer ce que peut apporter l'informatisation que ce que peut coûter la non-informatisation. Ce type d'évaluation est particulièrement délicat.

La multiplicité d'impacts de l'informatisation

Cependant, cette explication introduit l'idée convaincante et qui s'est imposée progressivement, que l'informatisation a une multiplicité d'impacts, trop longtemps réduits aux seuls gains de productivité (Jean-Louis Peaucelle, 1997, p. 41). Son influence est certainement au moins aussi importante sur les deux autres dimensions des caractéristiques d'usage indirectes que sont l'amélioration de la qualité des actions des utilisateurs et le raccourcissement des délais de production et de livraison. Les changements dans la qualité des biens et services produits, qui peuvent masquer des améliorations réelles de productivité, concernent l'augmentation du standard moyen exigé (par exemple pour la rédaction et la présentation des documents), la variété des produits voire la création de nouveaux produits, qui incorporent de plus en plus de fonctions informatiques. Le raccourcissement des délais peut porter sur l'approvisionnement et sur la fourniture de prestations en continu. Au niveau des processus de production, les effets de l'informatisation ne se limitent pas à l'automatisation totale ou partielle de certaines fonctions, mais concernent également la qualité des décisions prises (procédures de décision, quantité et qualité des informations mobilisées), la rapidité de réaction des entreprises à une évolution rapide de la demande des consommateurs, la fiabilité (notamment à travers les possibilités de traçabilité) et de façon plus générale une flexibilité potentiellement plus importante (Blaise Cronin, 1997, p. 10). Ainsi Jean Gadrey indique qu'il s'est produit "une progression souvent considérable, du volume et de la complexité des informations produites et traitées, en relation étroite avec la croissance de la complexité globale des services offerts : diversification des prestations offertes, complexification de l'environnement réglementaire, segmentation croissante des marchés, adjonction de nouvelles caractéristiques de services, de nouvelles garanties, individualisation des solutions, etc. Les livres de compte ont une progression de 40 % des lignes en dix ans, les dossiers médicaux et administratifs des malades à l'hôpital se sont fortement complexifiés, les stratégies de livraison "juste à temps" ont provoqué un gonflement du volume d'informations qui circulent entre les unités de production et le même phénomène existe dans les entreprises de service organisées sous forme de chaînes et de réseaux" (Jean Gadrey, 1996 A, p. 214). Cette multiplicité d'impacts de l'informatique s'est accentuée avec le passage de l'informatisation initiale de tâches bien identifiées, isolées et délimitées, au développement d'un système intégré d'informations de l'entreprise, de complexité croissante (cf. section I B) et portant sur "les mécanismes de coordination entre les agents, c'est à dire sur la manière dont s'articulent les tâches au sein du processus de travail, les fonctions au sein de la firme, les activités entre les firmes ou entre les secteurs" (Alain Rallet, 1997, p. 95-96).

Le problème est que ces impacts sont difficiles à mesurer directement. Si dans certains cas, on peut disposer d'indicateurs chiffrés significatifs (concernant les délais ou la disponibilité des services), c'est beaucoup moins évident pour évaluer la qualité d'un produit ou d'une décision. Par exemple une étude de l'OCDE sur l'impact des technologies de l'information dans le secteur des biens conditionnés conclut en indiquant que si certains aspects sont mesurables (niveaux des stocks, cas de rupture...) la plupart des changements sont qualitatifs (rapidité, commodité et variété) et défient toute mesure (OCDE, 1997 B, p. 212). De plus les modifications des caractéristiques des biens et des services rendent problématique la mesure de la production "en volume" et donc la perception de l'évolution de la productivité. C'est particulièrement net quand l'analyse est menée à un niveau désagrégé (poste de travail ou processus de production) où la production est mesurée à l'aide d'indicateurs physiques (unités d'œuvre). Quand l'analyse s'effectue à un niveau plus agrégé (entreprise, secteur, pays) ce phénomène est plus limité sans toutefois disparaître (cf. la sous-estimation de l'amélioration de la qualité des produits qui pourrait représenter jusqu'à un demi-point de croissance annuelle pour les Etats-Unis et qui aurait surévalué de plus de 1 % par an sur deux décennies l'indice des prix à la consommation, selon le rapport de M. Boskin, 1996), mais dans ce cas il est plus difficile d'isoler, parmi les gains, ceux qui proviennent de l'informatisation (Alain Rallet, 1997, p. 88).

Pour résoudre ces problèmes, on peut tenter d'apprécier indirectement les impacts de l'informatisation en examinant ses conséquences sur les performances des entreprises à partir de critères (rentabilité, compétitivité) pour lesquels on dispose d'indicateurs plus fiables. Il faut toutefois toujours mesurer l'informatisation de l'entreprise (l'autre terme de la relation). Or la mesure d'un stock de produits matériels et immatériels, opération toujours délicate, est rendue plus difficile par l'extension et l'intégration de l'informatisation qui a tendance à estomper les frontières entre technologies de l'information d'une part, automatisation industrielle et réseaux de télécommunications d'autre part (Alain Rallet, 1997, p. 88). Surtout, et c'est peut-être l'explication des résultats décevants des études empiriques de Claude Salzman et de Paul A. Strassmann citées précédemment, il peut s'opérer une redistribution des gains apportés par l'informatique au détriment de l'entreprise qui s'est informatisée (Erik Brynjolfsson, 1993). Jean-Louis Peaucelle prend l'exemple de la mensualisation des salaires à la fin des années soixante par versement sur un compte bancaire rendu possible par l'informatisation de cette opération, et qui a certes profité aux entreprises, mais également aux salariés, aux banques et à l'Etat (1997, p. 20). L.M. Hitt et E. Brynjolfsson (1996) ont tenté de

démontrer, par des méthodes économétriques, que l'informatisation aurait surtout profité aux consommateurs, notamment par la gratuité de nouveaux services et les baisses de prix. Les avantages pour les clients de l'informatisation peuvent être indirects : par exemple, dans le cas de l'entreprise Districast déjà mentionné, ce sont les entreprises clientes qui bénéficient de l'informatisation du distributeur, qui, en limitant le délai de livraison à 24 heures, leur permet de réduire fortement leurs stocks. A contrario pour Districast ceci peut être la source de coûts supplémentaires dus à la nécessité de gérer de plus petites commandes. (Catherine Palierne, 1998, p. 110). De façon plus générale, l'évolution des systèmes d'information vers l'aval et vers l'extérieur de l'entreprise (Marie-Christine Monnoyer-Longe, 1997, p. 120) accentue le partage des tâches entre les firmes à un point tel qu'il devient impossible d'identifier la contribution spécifique de chaque partie (Pascal Petit, 1998). C'est un constat similaire qu'effectue l'OCDE qui indique que "les technologies de l'information se caractérisent avant tout par les gains d'efficacité globale qu'elles permettent de réaliser en reliant différents acteurs dans des industries très diverses, et non par leur impact sur des entreprises données" (OCDE, 1997 B, p. 213).

La prise en compte du temps

La dernière explication, mais non la moins importante, de ce paradoxe est la prise en compte de la dimension temporelle, les effets de l'informatisation se manifestant avec retard (Erik Brynjolfsson, 1993). Au niveau microéconomique, une étude de H. Kivijärvi et T. Saarinen (1995) sur 200 entreprises finlandaises montre que ce sont les entreprises qui se sont informatisées depuis le plus longtemps, qui en tirent le plus de profit. Au niveau macroéconomique, la comparaison est effectuée avec l'électricité où il fallut attendre quarante ans pour que l'impact de l'électricité sur la croissance apparaisse dans les statistiques (P.A. David, 1991). Selon S. Roach, "la plupart des TI étant relativement récentes, les managers et les travailleurs informationnels en seraient seulement à un stade intermédiaire de la traditionnelle « courbe d'apprentissage » où l'expérience s'acquiert au prix d'erreurs et de certains gaspillages" (cité par Jean Gadrey, 1996 A, p. 151). Les explications de l'important décalage temporel observé sont dans les deux cas les obstacles à une diffusion généralisée de la nouvelle technologie (Chris Freeman, 1988, p. 54-59) et notamment la lenteur du processus d'apprentissage. La durée de ces apprentissages n'est pas très surprenante en raison du caractère ample et systémique des changements introduits (Pascal Petit, 1998, p. 29). Richard E. Walton (1989) distingue trois composantes d'apprentissage : l'apprentissage technologique (capacité d'utilisation de l'outil), l'apprentissage organisationnel (capacité d'adapter

l'organisation de l'entreprise aux nouvelles formes organisationnelles) et l'apprentissage d'affaires (capacité d'élaborer et de concevoir de nouveaux produits et services à partir du savoir-faire technologique et organisationnel). Or ces trois composantes ont une temporalité longue : l'apprentissage de la technologie par les utilisateurs ne s'effectue que progressivement dans un processus d'amélioration et de découvertes de nouveaux usages (Nathan Rosenberg, 1983), les changements organisationnels sont lents, coûteux et difficiles à mettre en œuvre (Eric Brousseau, 1997, p. 41) et la diffusion des technologies de l'information dans de nouveaux produits est encore plus longue que dans les procédés de production (Alain Rallet, 1997, p. 92). De plus, le processus d'apprentissage peut être considéré comme "un processus de rétro-alimentation entre ces trois composantes" (Ramon Salvador Valles, Lucas Van Wunnik, Félix Pineda, 1997, p. 139) et c'est seulement quand ce processus dynamique a permis d'atteindre un nouvel état d'équilibre que sont pleinement exploitées les potentialités de la nouvelle technologie. Or, la rapidité d'apparition de technologies nouvelles, introduisant des ruptures brutales dans les conditions d'utilisation des technologies de l'information, réactive constamment le cycle expérimentation, rationalisation et stabilisation (Alain Rallet, 1997, p. 86), et a, jusqu'à maintenant, empêché ce processus d'apprentissage de se dérouler complètement.

En conclusion et quelles que soient les explications apportées, il apparaît clairement qu'il est particulièrement difficile d'évaluer *ex ante* l'efficacité des technologies de l'information. Comme le note Frantz Rowe (1997, p. 24), "la productivité de l'informatique n'existe pas (...). On ne peut concevoir une performance de la technique au pied de la lettre, mais une performance appréhendée au niveau du n-uple activité-outil-utilisateur-commanditaire-organisation-client". Isoler au sein des technologies de l'information l'apport d'un logiciel particulier est encore plus problématique. Il en résulte que la valeur d'usage du logiciel peut difficilement servir de cadre théorique pour expliquer l'évolution des prix des logiciels.

3 - La théorie de la valeur-travail, une théorie inadaptée au cas des logiciels

La deuxième explication théorique des prix à partir d'une théorie de la valeur est la théorie de la valeur-travail de Karl Marx. Selon Marx, la valeur d'échange d'une marchandise est déterminée par la quantité de travail (mort et vivant) *socialement* nécessaire à sa production. Socialement signifie en moyenne vu le niveau de développement des forces productives de la société dans laquelle circulent les marchandises. En conséquence, le champ

d'application de cette théorie se limite aux produits reproductibles et Marx excluait explicitement le cas de ce que l'on peut appeler des biens tangibles non reproductibles (œuvre d'art originale par exemple).

Il nous semble que ce cadre théorique ne peut pas non plus s'appliquer aux logiciels. Concernant les logiciels sur mesure, qui sont un cas particulier de services non standardisés, le produit étant par définition significativement différent pour chaque prestation réalisée, la valeur d'échange ne peut être fondée que sur la quantité de travail nécessaire dans un cas particulier, ce qui est une autre façon d'expliquer la prédominance de la régie sur le forfait. A l'inverse les progiciels se caractérisent par leur très grande reproductibilité. Mais dans le cas des progiciels, comme dans celui plus général des biens intangibles, la quantité de travail pour réaliser un exemplaire supplémentaire (simple opération de copie) est dérisoire par rapport à la quantité de travail nécessaire pour développer l'original. De ce fait, la théorie de la valeur-travail n'est pas non plus susceptible d'expliquer les mouvements de prix pour ce type de biens. Il nous paraît plus judicieux d'analyser l'évolution des prix par l'existence d'une rente que pourrait s'approprier le titulaire des droits de propriété sur l'original. Cette rente est une forme particulière de rente différentielle, liée à l'avantage qu'un progiciel donné peut détenir par rapport à des produits concurrents, en termes de caractéristiques d'usage, cet avantage pouvant résulter de ses qualités propres mais également de l'importance de sa diffusion, en raison de la force des rendements croissants d'adoption. Ceci peut permettre d'expliquer les stratégies de prix bas, voire de gratuité, pratiquées pour s'imposer sur un marché et se retrouver ultérieurement dans une situation permettant de capter une rente importante. En effet, quand un progiciel a réussi à devenir hégémonique sur un segment de marché, processus qui peut être rapide et qui est difficilement réversible, ce progiciel peut être vendu à un prix élevé (rente différentielle très importante), ce qui se traduit par le très haut niveau de profitabilité que peuvent atteindre certains éditeurs de progiciels. Réciproquement, l'apparition d'un nouveau produit concurrent plus performant qui réussit à conquérir une part significative de marché, peut faire disparaître la rente différentielle dont bénéficiaient les produits précédents, expliquant les chutes brutales de prix constatées, qui peuvent aller jusqu'à la disparition d'entreprises très profitables précédemment (cf. certains éditeurs de progiciels pour micro-ordinateurs comme Aston-Thatte pour les bases de données ou WordPerfect pour les traitements de texte). Ce type de rente, que l'on peut assimiler à une rente technologique, existe dans tous les secteurs où les coûts de conception représentent une part importante des coûts (matériel informatique par exemple). Mais dans le cas des biens tangibles, il faut

également intégrer les coûts de l'activité de fabrication *stricto sensu* (la réalisation d'un exemplaire supplémentaire) qui sont insignifiants dans le cas des biens intangibles.

On peut également noter que la place prise par les logiciels dans la production a des conséquences essentielles pour la théorie de la valeur-travail concernant l'ensemble des marchandises. L'intuition de cette importance se trouve dans certains écrits de Marx, qui ne concernaient évidemment pas les logiciels, mais le *savoir abstrait*. Dans le "Fragment sur les machines", Marx défend l'idée que le savoir abstrait tend à devenir, en vertu précisément de son autonomie par rapport à la production, la principale force productive, reléguant dans une position marginale le travail parcellisé et répétitif. Pour Marx, il s'agit du savoir objectivé dans le capital fixe, qui s'est incarné dans le système automatique des machines. Marx pour désigner ces connaissances abstraites parle de *general intellect* (cerveau général), ce que Maximilien Rubel traduit par "puissance matérialisée du savoir". Paolo Virno (1992) propose d'élargir la notion de *general intellect* au-delà de la connaissance qui se matérialise dans le capital fixe, en y incluant aussi les formes de savoir incarnées dans le travail vivant qu'il appelle intellectualité de masse, "qualité et signe distinctif de toute la force de travail de l'époque post-fordiste (...) dans laquelle l'information, la communication jouent un rôle essentiel dans chaque repli du procès de production". Il nous semble plus significatif des transformations récentes de la production d'inclure dans le *general intellect* la part des connaissances codifiées non intégrées directement dans des machines mais qui peuvent circuler et être reproduites indépendamment de la force de travail et qui sont notamment présentes sous la forme de logiciels divers. Cette importance est analysée dans un autre cadre théorique par Herbert Simon (1992) qui parle des logiciels comme facteurs de production. La production de marchandises repose dans une proportion croissante sur des connaissances codifiées qui ne nécessitent quasiment pas de travail pour être reproduites ("librement" reproductibles). De ce fait, la mesure, dans le cadre du système capitaliste, de la valeur par la quantité de travail socialement nécessaire devient de plus en plus contradictoire avec une production qui s'appuie de plus en plus directement et principalement sur les connaissances. Comme le formule Manuel Castells (1998), "pour la première fois dans l'histoire, l'esprit humain est une force de production directe, et pas simplement un élément décisif du système de production".

Au terme de cette analyse, le logiciel apparaît ainsi comme un objet technologiquement complexe : il vise à apporter des solutions à des problèmes de plus en plus complexes – que dans certains cas il contribue à complexifier davantage – en étant partie prenante d'un

système, le système informatique, qui se caractérise également par une grande complexité. Il existe une dynamique puissante d'augmentation simultanée de la puissance des matériels et de la complexité des logiciels, basée sur la force des relations de complémentarité techniques et économiques entre ces deux activités.

Malgré sa jeunesse, le logiciel est marqué par un foisonnement d'innovations, qui concernent l'extension accélérée de ses domaines d'applications, la création de nouveaux produits, le renouvellement des méthodes et des techniques de production, et l'utilisation d'outils d'automatisation (génie logiciel). La forte dynamique innovatrice de l'économie du logiciel est confirmée par le fait que les innovations dans le domaine des logiciels concernent *tous* les types d'innovations mis en évidence par les économistes qui souhaitent appréhender l'ensemble des innovations existant dans l'économie. Par contre, les rythmes de diffusion très différenciés de ces innovations expliquent l'hétérogénéité technologique de l'économie du logiciel et le fait qu'elle semble caractérisée par une instabilité technologique permanente.

Sur un plan plus économique, le développement de logiciels sur mesure constitue une activité de services, alors que les progiciels sont des biens intangibles. Les progiciels possèdent "naturellement" des caractéristiques de biens collectifs des biens intangibles. De ce fait, pour rendre un progiciel appropriable, il est nécessaire de recourir à des dispositifs techniques (à l'efficacité limitée) et à des dispositifs juridiques. La variété de ces dispositifs, les problèmes que pose leur application aux progiciels, expliquent la diversité des statuts juridiques des progiciels et l'apparition de formes originales comme les licences publiques à la base des logiciels libres. Enfin, les logiciels se caractérisent par des mécanismes singuliers de détermination de leur prix. Les théories de la valeur ne semblent pas pouvoir s'appliquer aisément au cas des logiciels : il est particulièrement difficile d'appréhender la valeur d'usage d'un logiciel, ce qu'illustre particulièrement les débats sur l'appréciation des effets de l'informatique ("paradoxe de Solow"). Quant à la théorie de la valeur travail, elle ne peut s'appliquer à des biens qui, soit ne sont pas reproductibles (logiciels sur mesure), soit sont trop facilement reproductibles (progiciels).

Chapitre IV

L'IMPORTANCE DES LOGICIELS DANS L'ECONOMIE :

LA "LOGICALISATION" DE L'ACTIVITE ECONOMIQUE

Les logiciels font partie des technologies de l'information et de la communication. Une première interrogation concerne l'importance des logiciels et son évolution au sein de ces technologies. Elle nécessite d'étudier la place prise par ces technologies, leur convergence éventuelle permise par la numérisation des contenus, leur impact sur l'ensemble de l'économie et de la société. En effet, le succès d'un certain nombre d'expressions (société de l'information, société fondée sur les connaissances) semble indiquer que l'information et les connaissances jouent un rôle de plus en plus important. Nous nous interrogerons sur la réalité de ces phénomènes en étudiant le poids des biens et services informationnels, mais également le rôle que jouent l'information et les connaissances dans la production de l'ensemble des biens et des services, et dans le fonctionnement de l'économie. Au-delà, pour certains auteurs, les technologies de l'information et de la communication seraient à la base d'un nouveau paradigme économique, ce que nous étudierons à partir d'une confrontation entre les différentes significations que peut prendre la notion de paradigme appliquée à l'économie et les principales tendances observées dans le fonctionnement de l'économie.

La première section est consacrée à l'analyse du rôle croissant des technologies de l'information et de la communication et de ses implications. La deuxième section montre l'aspect de plus en plus important pris par les logiciels au sein des technologies de l'information et de la communication, ce qui nous amènera à parler d'une "logicalisation" de l'activité économique et sociale.

Section I - Le rôle croissant des technologies de l'information et de la communication

L'économie est de plus en plus fondée sur l'information et les connaissances, au niveau de la consommation et de la production (A), ce qui est tout à la fois la cause et la conséquence du développement des technologies de l'information et de la communication, dont certains auteurs pensent qu'elles sont à la base d'un nouveau paradigme technico-économique (B).

A - LE ROLE CROISSANT DE L'INFORMATION ET DES CONNAISSANCES

Définir l'information comme un processus de communication de connaissances (cf. chapitre I) rend indissociables les deux notions utilisées pour désigner l'évolution actuelle, d'économie de l'information (ou "économie intensive en information", Werner Schwuchow, 1997, p. 116) et d'économie fondée sur les connaissances. En effet, l'économie ne peut être de plus en plus basée sur les connaissances qu'à la condition que se développent significativement les informations, grâce notamment à de nouvelles technologies, qui permettent de faire circuler les connaissances entre les agents économiques. Réciproquement, l'ampleur et la diversité que prend l'information sous toutes ces formes et à tous les niveaux dans l'économie contemporaine, ne peuvent être comprises qu'en réponse à des besoins croissants d'acquisition de connaissances de la part de l'ensemble des agents économiques. On peut du reste constater que les auteurs qui étudient l'économie de l'information soulignent qu'il est nécessaire de prendre également en compte "la montée des connaissances et le rôle accru des savoirs" (Pascal Petit, 1998, p. 16), et que les auteurs qui traitent de l'économie fondée sur les connaissances intègrent nécessairement la place prise par l'information. C'est pour cette raison que nous analyserons simultanément le rôle croissant de l'information et des connaissances.

Celui-ci peut être appréhendé sous deux aspects complémentaires : d'une part, le contenu informatif et cognitif de plus en plus important des biens et les services consommés par les ménages ou les entreprises (1), d'autre part, la place croissante de l'information et des connaissances dans la production de l'ensemble des biens et services (2). L'importance croissante de l'information et des connaissances a été confirmée par des études empiriques, dont certains auteurs concluent à l'avènement d'une nouvelle société (3).

1 - Les biens et les services ont un contenu informatif et cognitif de plus en plus important

On peut distinguer deux phénomènes. D'une part, les biens et les services dont la composante informative et cognitive est déterminante occupent une place croissante dans l'économie (a). D'autre part, l'ensemble des biens et des services a une dimension informationnelle qui est de plus en plus importante (b).

a - La place croissante des biens et services dont la composante informative et cognitive est déterminante

Les activités qui consistent à fournir des biens et services d'informations sont désignées, avec des contenus parfois légèrement différents, par les notions d'industrie de l'information, de secteur de l'information ou d'activités informationnelles. Le secteur, dans lequel sont intégrées les technologies de l'information et de la communication, est celui qui connaît la croissance la plus rapide dans l'économie (Werner Schwuchow, 1997, p. 114). En termes de biens, il comprend les biens électroniques (ordinateurs, équipements de télécommunication...) mais également les biens intangibles reproductibles sous des formes de plus en plus diverses. En termes de services, il englobe les services de télécommunications et les services de fournitures d'informations (depuis les bibliothèques jusqu'aux services télématiques). Certains auteurs considèrent que les services (aux ménages ou aux entreprises) dont l'activité principale est basée sur du traitement d'informations et de la transmission de connaissances en font également partie (sociétés de conseil, enseignement, voire les banques et les assurances) et relie cette croissance à la tertiarisation de l'activité économique (Pascal Petit, 1998). En se limitant à une conception plus restreinte des secteurs informationnels, on constate que l'essentiel de ces activités est tourné vers les entreprises : C. Jonscher (1984) indique qu'aux Etats-Unis en 1983, la consommation de services d'information ne représente que 7,8% de la consommation finale totale, contre 37,7 % des consommations intermédiaires de l'industrie, même s'il faut ajouter que la consommation des ménages en activités informationnelles croît rapidement sur la période récente.

b - L'importance croissante de la dimension informationnelle dans l'ensemble des biens et services

En plus de la croissance des biens et services informationnels, l'ensemble des biens et des services comprennent une dimension informationnelle de plus en plus importante Celle-ci résulte de deux phénomènes : la généralisation de l'utilisation de l'électronique, et le fait que

les prestations de services ou les ventes de biens s'accompagnent de plus en plus de la fourniture d'une quantité croissante d'informations.

Concernant le premier point, on peut noter d'une part que de plus en plus de biens intègrent des composants électroniques de plus en plus puissants : selon Jean-Benoît Zimmermann, la tendance est au doublement, tous les deux ans, du volume de code intégré dans les biens de consommation¹ (1998 A, p. 131), une automobile contient actuellement plus de puissance informatique qu'un micro-ordinateur (Daniel Kaplan, 1999), au Japon, les "bâtiments intelligents" ont un équipement électronique qui peut représenter jusqu'à 25 % du coût total (Chris Freeman, 1995, p. 29). Par ailleurs, de plus en plus de services sont réalisés sous forme électronique. Ces transformations permettent d'améliorer les caractéristiques techniques et les caractéristiques d'usage directes des biens et des services, par leurs possibilités de traiter une masse croissante d'informations saisies par l'utilisateur ou captées sur l'environnement (programmabilité, réactivité...).

Le deuxième point (cession conjointe d'informations) permet principalement d'améliorer les caractéristiques d'usages indirectes des biens et des services en aidant l'utilisateur à exploiter les potentialités d'un bien (conseils pour l'installation, l'intégration, l'utilisation, l'entretien), en améliorant ses compétences (notamment dans le cas de prestations coproduites) et en lui fournissant des informations complémentaires (par exemple, les clients de Federal Express peuvent savoir en temps réel où se trouvent les marchandises transportées). Ces aspects, qui sont une des dimensions de la compétitivité hors prix, deviennent de plus en plus important pour les consommateurs (particuliers ou entreprises). Eric Brousseau (1993, p. 286) cite l'exemple de Dupont de Nemours qui permet aux entreprises clientes de rapatrier des fichiers comprenant l'ensemble des caractéristiques des produits qu'elles ont achetés, ce qui leur permet de les intégrer directement dans leur système de conception assistée par ordinateur (CAO).

¹ Actuellement un rasoir intègre déjà deux kilo-octets de logiciel, une télévision 500 kilo-octets.

2 - La place croissante de l'information et des connaissances dans la production de l'ensemble des biens et services

Elle peut être appréhendée au niveau microéconomique (utilisation de l'information comme ressource économique critique pour les entreprises et les organisations) et à un niveau global (importance de l'information dans le fonctionnement de l'économie actuelle).

a - Au niveau microéconomique, l'importance des investissements immatériels et des systèmes d'information

Les informations et les connaissances sont un input de plus en plus important dans les produits des économies, certains auteurs allant jusqu'à considérer que l'information est devenue "la composante principale des biens comme des services" (Faïz Gallouj, 1994, p. 98). Cette place croissante de l'information "qui sert (...) de biens intermédiaires et d'investissement pour la sphère productive" (Michel Catinat, 1998, p. 44) correspond à la croissance des biens et services informationnels, dont nous avons souligné qu'ils étaient principalement utilisés par les entreprises (cf. 1 a).

Le champ des services informationnels est vaste. Jacques De Bandt (1995) repère parmi les services informationnels ceux liés à la relation de marché (étude de marché, mercatique), à la production (maintenance, organisation de la production, essai), à la stratégie de l'entreprise (conseils, formation, financement, recherche). Les achats de services informationnels sont considérés comme des consommations intermédiaires ou comme des investissements (immatériels), la frontière entre les deux n'étant pas toujours évidente.

L'investissement immatériel est défini par le Comité national de l'information statistique (CNIS) comme "une dépense qui, bien qu'inscrite en charge d'exploitation, développe la capacité de production et valorise l'entreprise en s'accumulant, sous la forme d'un capital amortissable, sur une production future et en constituant une valeur patrimoniale, cédable sur le marché" (Christine Afriat, Pierre Caspar, 1988, p. 21). L'appréhension des investissements immatériels, notamment sur un plan comptable, est délicate en raison de deux particularités propres à ces investissements : la part de production interne est beaucoup plus importante que pour les investissements matériels, d'où une plus grande difficulté pour mesurer le coût de ces investissements ; la relation entre l'investissement immatériel et la capacité de produire et de vendre est le plus souvent indirecte, ce qui complexifie l'analyse de leur rentabilité (Christine Afriat, 1992, p.99). Les investissements immatériels englobent des dépenses de prospection, de recherche-développement, d'études, de publicité mais également certains investissements

en commercialisation, en formation, en organisation et en amélioration du fonctionnement interne (Christine Afriat, Pierre Caspar, 1988, p. 17-18). La part des investissements immatériels a augmenté d'environ 50 % (en termes de points de PIB en une décennie), passant de 2,6 % à 3,7 % du PIB entre le milieu des années soixante-dix et le milieu des années quatre-vingt, alors que dans le même temps l'investissement matériel passait de 17,6 % du PIB à 15 % (OCDE, 1992, p. 113). J.W. Kendrick (1994) estime que le stock de capital intangible (éducation, formation, recherche-développement, santé) est devenu équivalent au stock de structures et équipement (principale sous-catégorie du stock de capital tangible) vers 1973 et qu'il est aujourd'hui largement dominant (cité par Dominique Foray et Bengt Ake Lundvall, 1997, p. 18).

La maîtrise de l'information et des connaissances est devenue un facteur décisif de compétitivité des entreprises. Jean-Claude Tarondeau (1998, p. 72) recense différentes études empiriques qui montrent que les actifs intangibles sont plus importants que les actifs tangibles dans l'explication des performances des firmes. De même, l'OCDE estime que "ce qui détermine la réussite des entreprises (...) dépend plus que jamais de leur efficacité à rassembler et à utiliser des connaissances " (OCDE, 1996, p. 14). Dans un régime d'innovation intense, il est nécessaire de reconstituer en permanence le capital de connaissances (Armand Hatchuel, 1996). Ikujiro Nonaka et Hirotaka Takeuchi (1997) estiment que le succès de certaines entreprises japonaises provient principalement de leur capacité à créer de nouvelles connaissances.

Selon Paul Davenport "les économies des nations marchandes évoluent rapidement vers un modèle mondial fondé sur la connaissance, qui exige une évolution rapide des systèmes d'information" (1997, p. 11). On peut définir le système d'informations comme "l'ensemble des informations formelles circulant dans l'entreprise ainsi que les procédures et les moyens nécessaires pour les définir, rechercher, formaliser, conserver et distribuer" (Christiane Dumoulin, 1986). Un "système d'information est un langage de communication de l'organisation, construit consciemment pour représenter, de manière fiable et objective, rapidement et économiquement, certains aspects de son activité passée ou à venir" (Jean-Louis Peaucelle, 1981, p. 24-25).

Le système d'information prend de plus en plus d'importance quand les besoins informationnels sont de moins en moins ceux que suppose la réalisation d'une tâche selon une démarche très "fordienne", mais ceux qu'implique la gestion de processus et d'événements, au

sein d'un système diversifié d'interactions internes à l'entreprise pour la réalisation complète d'un produit ou d'un service (Marie-Christine Monnoyer-Longé, 1997, p. 119). Quand, de plus, la concurrence entre les entreprises se transforme d'une "guerre de position", où la stratégie peut se permettre d'être statique, à une "guerre de mouvement" où l'essence de la stratégie d'une entreprise est la dynamique de son comportement (G. Stalk, P. Evans, L. E. Shulman, 1992, p. 62), le système d'information devient une composante essentielle des entreprises.

En même temps, les systèmes d'informations se transforment, dans une société où la fonction commerciale de subordonnée à la fonction productive tend à devenir l'activité première (Christine Afriat, 1992, p. 46-47). Ils doivent de plus en plus être orientés vers l'aval et vers l'extérieur de l'entreprise (Marie-Christine Monnoyer-Longé, 1997, p. 120). Godefroy Dang Nguyen montre comment les systèmes de réservation aérienne (SIR), projets gigantesques qui ont mis plus de trente ans pour être développés, ont été durant les décennies soixante-dix et quatre-vingt, un élément essentiel de la compétitivité des deux leaders du transport aérien américain, American Airlines et United Airlines (1995, p. 346). La richesse des interactions de l'entreprise avec ses partenaires, fournisseurs et sous-traitants, prend également de plus en plus d'importance (Anne Mayère, 1997, p. 207). Une "firme-réseau" comme Benetton, qui sous-traite 80 % de sa production à 350 petits producteurs et qui détient un réseau de 4600 magasins franchisés, n'a pu connaître le succès que grâce à l'existence d'un système d'information très performant reliant les magasins, le siège central et son informatique de gestion, et les fournisseurs (Godefroy Dang Nguyen, 1995, p. 199-200).

L'aboutissement de ces processus est, au moins dans l'industrie, la production en "flux tendus", qui associe dans un seul réseau d'information, un réseau de fournisseurs, assembleurs, transporteurs et distributeurs. Cette forme de production vise à réduire les stocks et les surcapacités (c'est à dire à substituer de l'information au capital) mais permet également d'individualiser la production pour répondre à la diversité et aux évolutions des demandes du marché (Daniel Kaplan, 1999). Pour rendre compte du fait que, de plus en plus, une entreprise "doit centrer les moyens qu'elle met en œuvre pour atteindre ses objectifs sur ses ressources en information", a été développée la notion "d'organisation basée sur l'information" (OBI), initialement introduite par Peter F. Drucker (Paricia D. Fletcher, Lester Diamond, 1997, p. 170). Le développement récent du management des connaissances (*knowledge management* ou *KM*) et de l'intelligence économique, témoigne également de cette évolution.

b - Au niveau du fonctionnement de l'économie

Les tendances analysées précédemment sur un plan microéconomique peuvent être extrapolées au niveau du fonctionnement de l'ensemble de l'économie. Logiquement la coordination entre des entreprises utilisant de plus en plus d'informations pour produire des biens et des services dont la dimension informative est de plus en plus importante, requiert des flux d'informations croissants. Dans certains cas (mode de fonctionnement en réseaux), "le marché a été remplacé par des flux d'information harmonisés en tant que mécanisme premier de coordination" (Sandra Braman, 1997, p. 89). Il faut toutefois préciser que la création de structures résiliables ne signifie pas nécessairement la disparition des relations marchandes : Eric Brousseau cite l'exemple, aux Etats-Unis, de la mise en œuvre en 1986 d'un "partenariat" entre de grandes entreprises de la distribution, de la confection, du textile et de la chimie, dont l'objectif était de gérer en flux tendus l'ensemble de la filière textile, par le développement de normes en matière d'étiquetage (code-barre) des produits et des emballages, des documents commerciaux (bons de commandes, facture, avis de paiement, etc.), ainsi que des logiciels de gestion des stocks et des commandes (1993, p. 302-303). Mais surtout, même quand le marché continue à jouer un rôle moteur dans les échanges, le fonctionnement des marchés ne s'appuie plus seulement sur les prix (pour lesquels l'information les concernant s'est fortement accrue), mais sur tout un ensemble d'informations concernant les qualités "cachées" du produit, ce que reflète l'importance prise par la "compétitivité hors prix" (Pascal Petit, 1998, p. 20-21). L'extension des marchés et des combinaisons productives à l'échelle planétaire (mondialisation), la variété des produits, leur disponibilité immédiate sans stock, la multiplication des activités bancaires et financières "en temps réel" complètent "cette image d'une sphère virtuelle où les flux d'information propulsent l'économie mondiale" (idem, p. 21). Ces flux d'informations jouent un rôle d'autant plus décisif que l'évolution du cadre général d'exercice des activités économiques (mutations techniques, globalisation des activités, libéralisation de très nombreux marchés) a tendance à accroître l'imprédictibilité concernant les évolutions quantitatives et qualitatives de la demande et des coûts, l'émergence de nouveaux produits ou process, l'arrivée de nouveaux concurrents (Eric Brousseau, 1997, p. 50).

Cette importance croissante de l'information sur le plan macroéconomique a fait l'objet d'un certain nombre d'études empiriques, confirmant selon certains auteurs l'avènement d'une nouvelle société.

3 - Justifications empiriques et avènement d'une nouvelle société

a - Les études empiriques

Les premières études empiriques réalisées aux Etats-Unis par Fritz Machlup (1962) portaient sur les "industries du savoir". Ce secteur, spécifiquement dédié à la production et au traitement de l'information, était défini comme comprenant des activités de communication, éducation et média, des machines informationnelles, des services informationnels et d'autres activités. Les industries du savoir représentaient environ 29 % du PNB aux Etats-Unis en 1958 et 49 % en 1980 (Fritz Machlup, 1984).

La deuxième étude d'envergure est le travail mené par Marc Uri Porat (1977) sur l'économie de l'information. Il distingue au sein de celle-ci, deux secteurs d'informations : le secteur d'information primaire, chargé de créer et de fournir au marché les produits et services d'information, et le secteur d'information secondaire qui regroupe les services d'information produits pour l'usage interne des entreprises et des administrations. La démarche, pour évaluer l'importance de l'économie de l'information, consiste à utiliser les données dont on dispose sur l'emploi et les professions, pour classer les emplois en fonction de leur contenu de recherche-développement, de savoir ou d'information. Sur cette base, les activités de l'information représentaient 47 % du PNB aux Etats-Unis en 1967. En reprenant les indicateurs de M.U. Porat, M. Castells et Y. Ayoama (1994) estiment qu'au début des années quatre-vingt-dix, 45 % des emplois non agricoles dans les pays du G7 consistaient à "traiter" de l'information.

On peut également citer l'étude de C. Jonscher (1984), qui montre qu'une part de plus en plus importante des inputs de l'industrie et des services "non-informationnels" est composée de flux de services informationnels. Pour les Etats-Unis, la contribution nette des secteurs informationnels (banques, édition, bases de données...) à la valeur des biens du secteur de la production vendus aux consommateurs finaux, est passée de 27,8% en 1947 à 42,7% en 1983.

b - L'avènement d'une nouvelle société ?

Ces études confortent l'idée de l'avènement d'une société post-industrielle, formulée initialement en 1960 par Daniel Bell, comme stade inévitable du développement des sociétés, qui prenait la suite des étapes agricole et industrielle. Les conceptions de cette société sont diverses. On peut citer la notion de "troisième vague" d'Alvin Toffler (1990), qui désigne le passage à une économie basée sur l'information, où la connaissance est la source par excellence du pouvoir. Peter Drucker (1993) va jusqu'à affirmer que le trait distinctif de la

nouvelle société est que la connaissance n'est pas seulement une nouvelle ressource qui s'ajoute aux facteurs de production traditionnels, mais la seule ressource qui ait une signification réelle aujourd'hui, d'où le rôle central dans la production des "travailleurs de connaissance". C'est une problématique voisine, synthétisée dans le concept "d'entreprise intelligente", qu'expose James Quinn (1993) : le pouvoir économique de l'entreprise moderne tient plus dans ses capacités intellectuelles et de service que dans ses actifs matériels ; la valeur de la plupart des produits et services dépend fondamentalement de la façon dont sont développés les "actifs intangibles basés sur les connaissances", tels que le savoir-faire technologique, le design des produits, la présentation marketing, la compréhension des clients, la création personnelle et l'innovation.

En parallèle avec ces approches essentiellement managériales, les nouvelles théories de la croissance (P. Romer, 1994, G.M. Grossman et E. Helpman, 1994) intègrent le rôle du savoir et de la technologie comme moteur de la croissance économique. L'OCDE parle de "l'économie fondée sur le savoir" (1996) pour désigner des économies "qui reposent directement sur la production, la diffusion et l'utilisation du savoir et de l'information" (p. 7), et où "plus de 50 % du PIB des grandes économies de l'OCDE reposent maintenant sur le savoir" (p. 9). Dominique Foray et Bengt Ake Lundvall (1997) utilisent la notion "d'économie fondée sur la connaissance" pour désigner "la formation d'un nouveau régime économique, dans lequel les agents doivent se préparer à poursuivre des activités où les processus de production, de transformation, codification et mémorisation des connaissances seront déterminants" (p. 21). Selon ces auteurs, l'avènement de l'économie fondée sur la connaissance s'exprime moins par l'expansion continue d'un secteur spécialisé que par la prolifération d'activités intensives en connaissance dans tous les secteurs de l'économie (p. 19). A partir d'une caractérisation de l'économie de l'information et des changements structurels interdépendants qu'elle comporte, Pascal Petit esquisse les traits d'un nouveau régime de croissance d'inspiration régulationniste (1998, 1999).

Par ailleurs, mais en lien avec les approches précédentes, le concept de société de l'information, apparu dès le début des années quatre-vingt², est devenu un thème central de débat ces dernières années, notamment avec le développement d'Internet et du World Wide

² Il faut toutefois noter que, au Japon, dès le début des années soixante-dix étaient utilisées les expressions d' "économie de l'information" et de "société de l'information" (OCDE, 1993 B, p. 21).

Web. L'incertitude sur les caractéristiques et les rouages de la société de l'information, qui va de pair avec la certitude de son émergence et de ses potentialités (Michel Catinat, 1998, p. 37), entraîne une multiplication des rapports des pouvoirs publics sur les mesures à prendre. On peut citer les conclusions de la présidence de la Conférence ministérielle du G7 sur la société de l'information (25-26 février 1995), les travaux de l'OCDE lors de la Session spéciale du Comité de la politique de l'information, de l'informatique et des communications sur les infrastructures de l'information (3-4 avril 1995), le rapport de Al Gore et Ronald H. Brown (1995) aux Etats-Unis. Pour la communauté européenne, la société de l'information a donné lieu au Livre Blanc "Croissance, Compétitivité, Emploi" de Jacques Delors en 1993, au rapport Bangemann "l'Europe et la société de l'information planétaire" en 1994, et à un rapport d'un comité d'experts présidé par Luc Soete "Construire la société européenne de l'information pour tous" en 1996.

La société de l'information est également un thème privilégié de création de forums de discussion et de clubs de réflexion, comme par exemple en France le Club de l'Arche qui réunit des hommes d'entreprises, des consultants et des chercheurs. Dans un texte désigné comme le "Manifeste de l'Arche", au titre évocateur ("L'aube d'un monde nouveau"), est effectuée une comparaison entre l'ère de la mécanisation et l'ère de l'informatisation "qui porte probablement en elle le germe d'une future société d'abondance" (p. 3). Entre ces deux phases sont mis en parallèle, la voiture personnelle et le micro-ordinateur, les autoroutes de béton et les autoroutes électroniques avec des conséquences comparables sur la révolution radicale du mode de vie de chacun : invention " de nouvelles façons d'aller au bureau, au supermarché, au cinéma, non plus en voiture, mais "en modem" (...). Les nouvelles technologies, en permettant un accès élargi à la connaissance, en inventant une nouvelle génération de loisirs, ou en permettant de "travailler" chez soi ou "au pays" (...) peuvent s'imposer comme un vecteur de sortie de crise" (idem).

Ce type d'approche passe d'une définition extensive de la société de l'information, à savoir une société où l'information joue un rôle déterminant, à une définition plus restrictive comme société des technologies de l'information.

B - LES TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION A LA BASE D'UN NOUVEAU PARADIGME ?

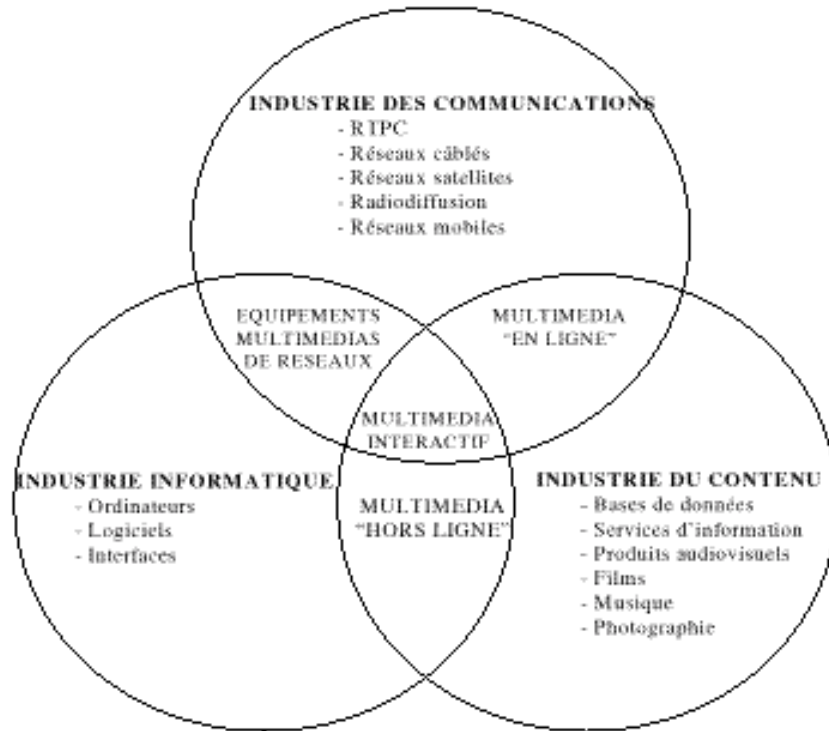
A partir de l'importance prise par les technologies de l'information et de la communication (1), certains auteurs concluent à l'avènement d'un nouveau paradigme qualifié de technologique, technico-économique ou de socio-économique (2). Si ces technologies jouent incontestablement un rôle de plus en plus essentiel dans les évolutions actuelles, il nous semble toutefois qu'il n'en découle pas nécessairement qu'un seul paradigme devienne dominant, et que, au contraire coexistent durablement plusieurs "paradigmes" différents (3).

1 - L'importance des technologies de l'information et de la communication

a - Le constat statistique : une croissance très rapide

Si la notion de technologies de l'information et de la communication est largement utilisée, le champ couvert par ces technologies est difficile à délimiter, ce qui a des conséquences importantes sur l'interprétation des évaluations statistiques existantes. Si on englobe tout ce qui est, utilise, incorpore, transforme, transporte et crée de l'information, l'ensemble des biens et services sont concernés. Par exemple, retenir tous les produits qui contiennent un microprocesseur conduirait tendanciellement à intégrer tous les produits manufacturés. A l'inverse retenir comme critère la numérisation de l'information, dont l'importance dans la convergence fonctionnelle de ces technologies est décisive (cf. supra, b) apparaît comme trop limitatif en excluant la part des télécommunications et la majorité de l'audiovisuel qui sont encore analogiques, et toute la sphère papier (anciennes technologies de l'information et de la communication). De ce fait la plupart des définitions des technologies de l'information et de la communication sont des définitions en extension, basées sur une nomenclature d'activités et de produits, élaborée de façon pragmatique (Secrétariat d'Etat à l'Industrie, 1999, p. 9). Cette nomenclature comprend trois composantes principales, l'industrie informatique, l'industrie des communications et l'industrie du contenu, dont l'articulation est représentée dans le schéma suivant, mais dont les frontières sont difficiles à délimiter : par exemple faut-il inclure la Poste dans l'industrie des communications, quelles sont les branches (édition, presse, audiovisuel...) qui font partie de l'industrie du contenu ?

Schéma VIII
Représentation du champ des technologies de l'information et de la communication



Source : OCDE (1997 A, p. 14)

Au centre des technologies de l'information et de la communication, se trouve le multimédia, combinaison d'informations de nature diverse (au moins textes, images et sons) et leur présentation sous des formes et sur des supports variés, avec des possibilités plus ou moins étendues d'interactivité contrôlée par un dialogue avec l'utilisateur. A partir des industries de "contenu" le multimédia n'a pu se développer de façon spectaculaire que grâce aux progrès de l'informatique et, pour la fourniture de produits "en ligne", des télécommunications.

Certes les problèmes de délimitation ne sont pas propres aux technologies de l'information et de la communication. Déjà en 1933, Edward H. Chamberlin notait que "le monde économique peut être considéré comme un réseau de marchés et de secteurs interdépendants, dont les frontières sont choisies de façon à rendre la représentation commode, mais qui sont essentiellement arbitraires". Toutefois les technologies de l'information et de la communication présentent des spécificités qui rendent leur impact

économique très difficile à mesurer : leur caractère d'emblée *mondial* alors que les conventions statistiques sont différentes selon les pays, l'importante évaluation *qualitative* qui exige de modifier les techniques statistiques traditionnelles, et le fait que l'effet le plus important de ces technologies se manifeste dans les *changements économiques indirects* induits par leur utilisation (OCDE, 1997 B, p. 6-7).

Pour les technologies de l'information et de la communication, le marché mondial était évalué à 1 227 milliards de dollars en 1994 (cf. tableau X), ce qui représente entre 5 et 6 % du PIB mondial. Si ce marché est légèrement inférieur au seul marché de l'équipement automobile neuf, il connaît une croissance beaucoup plus élevée que l'industrie manufacturière. Il faut noter également que les services représentaient 70 % du total en 1994, et que leur part augmente d'année en année au détriment des équipements (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 35-36).

Tableau X
Le marché mondial des technologies de l'information et de la communication en 1994
(en milliards de dollars courants)

	Télécoms	Informatique	Audiovisuel	Total
Equipements	95	198	86	379
Services (*)	430	233	185	848
Total	525	431	271	1 227
Etats-Unis	171	185	99	455
Europe	155	125	78	358
Reste du monde	199	121	94	414

(*) Y compris les progiciels pour l'informatique

*Source : OCDE / OMSYC
in Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 36*

L'UIT pour la même année estimait le marché des technologies de l'information et de la communication à 1 425 milliards de dollars (Eurostat, 1995 A, p. 4).

Pour la France, à partir d'une définition légèrement différente des technologies de l'information et de la communication englobant pour les "industries du contenu", l'audiovisuel

(télévision, cinéma, vidéo et radio) mais également le domaine de l'écrit (édition et presse), celles-ci représentaient 4,6 % du PIB (Secrétariat d'Etat à l'Industrie, 1999, p. 116).

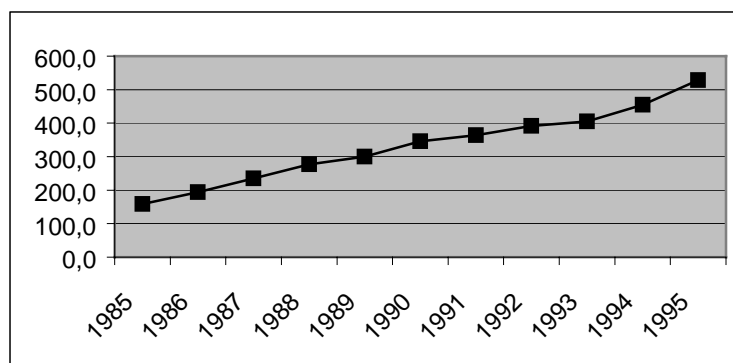
Au Japon les technologies de l'information, définies comme étant les télécommunications et l'informatique, sont depuis 1995 le premier secteur économique du pays. Elles représentent 10,3 % de la production industrielle et ont créé 760 000 emplois sur les 6 dernières années (Eurostaf, 1997 A, p.79).

Au sein des technologies de l'information et de la communication, ce sont les technologies de l'information, comprenant les ordinateurs, les équipements de communications de données, les logiciels et les services informatiques, qui connaissent la croissance la plus rapide.

Tableau XI
L'évolution du marché mondial des technologies de l'information
(en milliards de dollars courants)

Année	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
Total des T.I.	158,3	193,5	235,1	276,6	300,3	345,4	364,0	391,6	405,1	455,0	527,9
Taux de croissance		22%	21%	18%	9%	15%	5%	8%	3%	12%	16%

Graphique I



Source : calculs et représentation effectués d'après des données IDC, citées in OCDE, 1997 B, p. 251.

Le taux de croissance annuel moyen a été de 12,8 % sur la période. On constate que la croissance très forte durant les années quatre-vingt (phase d'équipement à outrance) s'affaiblit au début des années quatre-vingt-dix sous la conjonction de plusieurs facteurs : érosion de la croissance des budgets informatiques sur fond de crise économique, marché de renouvellement plus que de premier équipement, forte baisse des prix liée au passage des

systèmes propriétaires aux systèmes ouverts dans le *hardware* et des logiciels sur mesure aux progiciels dans le *software* (Eurostaf, 1995, p.9). Toutefois cette phase de croissance plus faible, qui a eu des effets importants sur certaines entreprises du secteur habituées à une croissance très élevée (par exemple IBM voit ses effectifs passer de 407 000 salariés en 1986 à 220 000 en 1994) semble n'avoir été qu'une parenthèse. La reprise d'une croissance forte dans le secteur informatique, perceptible depuis le milieu des années quatre-vingt-dix, semble s'accélérer actuellement en raison de facteurs très conjoncturels (passage à l'Euro, bogue de l'an 2000) mais également plus structurels (croissance exponentielle d'Internet, meilleures perspectives économiques). Par exemple, aux Etats-Unis la part des technologies de l'information est passée de 30 % du budget d'équipement des entreprises en 1990 à 40 % en 1996 selon l'OCDE (1997 B, p. 13), Patricia Fletcher et Lester Diamond estimant que les investissements des entreprises dans les technologies de l'information ont même dépassé depuis 1991 les sommes consacrées aux équipements traditionnels, bâtiments, machines et autres actifs industriels (1997, p. 175).

Enfin il est important de noter que la part grandissante prise par les technologies de l'information et de la communication provient essentiellement de sa composante services (au sens large y compris les progiciels), la fabrication de matériel d'information et de communication représentant généralement 2 % du PIB et un peu plus de 1 % de l'emploi, chiffres qui n'ont guère évolué depuis les années 70 (OCDE, 1997 B, p. 6). Les investissements matériels sous formes d'ordinateurs et de moyens de télécommunications gardent une place modeste estimée à 11,7 % du stock net de capital fixe productif (seulement 2 % pour les ordinateurs et les périphériques) aux Etats-Unis en 1993 (Pascal Petit, 1998, p. 18). Ceci s'explique par la forte baisse des prix de ces matériels qui est un des facteurs de la rapidité de diffusion des technologies de l'information et de la communication.

b - Les facteurs de la rapidité de diffusion

Les deux facteurs principaux sont la numérisation qui est à la base d'une convergence fonctionnelle des technologies de l'information et de la communication (i) et la formidable baisse des prix notamment des composants électroniques (ii). L'interaction entre ces deux facteurs entretient un cercle vertueux qui assure la diffusion rapide des technologies de l'information et de la communication (iii).

i - La numérisation à la base d'une convergence fonctionnelle des technologies de l'information et de la communication

Un des facteurs explicatifs de la rapidité de diffusion actuelle, et qui permet d'anticiper une accélération prévisible de la pénétration des technologies de l'information et de la communication, est la numérisation. La numérisation consiste à exprimer sous une forme numérique discrète une information analogique continue, ce qui permet d'intégrer, de transmettre, de traiter, voire d'analyser des contenus de nature différente, traduits en des signaux techniquement identiques. La technologie numérique permet de réduire à la fois les coûts de reproduction et les coûts de distribution, qui sont des coûts distincts mais souvent confondus (Carl Shapiro, Hal R. Varian, 1999, p. 81). On peut considérer que la numérisation est un langage ou plus précisément un métalangage (langage des langages) si on définit le langage comme un système de signes remplissant une fonction de communication (Jean-Louis Caccamo, 1996, p. 44-45). La numérisation du signal se substitue progressivement au traitement analogique dans l'ensemble des secteurs des technologies de l'information et de la communication. Dans l'informatique, le choix de la technologie numérique a été effectué dès les débuts de l'informatique par l'ensemble des constructeurs, après la brève coexistence dans les années cinquante de deux familles d'ordinateurs analogiques et numériques. Dans les télécommunications, où l'analogique a régné en maître pendant trente ans, le numérique se généralise rapidement : la numérisation des réseaux dans les pays de l'OCDE est passée de 49 % en 1991 des lignes principales à près de 80 % en 1995 (OCDE, 1997 A, p. 11). Si la numérisation de la télédiffusion est plus récente, elle semble inéluctable.

La numérisation crée progressivement les conditions d'une connectivité généralisée dans l'environnement professionnel et domestique (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 35). Elle provoque une convergence fonctionnelle dans un espace numérique intégré entre les différentes filières informationnelles (télévision, radio, téléphone, informatique), au départ différenciées selon la nature particulière d'un signal technique de traitement, à la base d'un "métier" et d'une structure industrielle spécifiques (Jean-Louis Caccamo, 1996, p. 76). Les secteurs centrés initialement sur la communication incorporent une part croissante de traitement de l'information, avec les possibilités qu'offrent la numérisation en terme de compression, de cryptage, d'intégration de données de nature très diverses, alors que l'ordinateur dépasse sa fonction originelle de traitement de l'information pour devenir également un outil de communication.

Cette convergence fonctionnelle, entérinée sur un plan législatif aux Etats-Unis par le vote d'une loi le 1-2-96 qui prévoit la suppression de tout cloisonnement sectoriel à partir du 31-3-99, a contribué à des modifications importantes comme l'ouverture et déstructuration des réseaux informatiques et de télécommunications (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 58). Elle n'implique pas toutefois une fusion des marchés aussi différents que sont les télécommunications, l'édition, l'informatique ou l'audiovisuel. En effet, en anticipant une convergence des usages, des entreprises ont tenté de se redéployer dans d'autres secteurs (par exemple IBM vers les télécommunications et ATT vers l'informatique). Ces tentatives se sont la plupart du temps soldé par des échecs et la tendance actuelle consiste plutôt à se recentrer sur son métier de base (Godefroy Dang Nguyen, Denis Phan, 1997, p. 186-187, Gérard Dréan, 1996 A, p. 350)³, et c'est pour cette raison qu'un rapport récent préfère parler d'*infosphère* plutôt que de convergence.

S'il est vraisemblable que les industries numériques resteront différenciées au niveau des applications avec une diversification des champs de valorisation, elles s'homogénéisent au niveau des composants. Que la numérisation concerne des produits avec "l'information comme bien de consommation" (téléphone, radio, télévision, matériel Hi-Fi, jeux, informatique domestique) ou des processus avec "l'information comme bien intermédiaire" (Machines Outils à Commande Numérique, informatique d'entreprise), elle repose en amont des filières sur l'industrie des semi-conducteurs, principalement les mémoires et les microprocesseurs (Jean-Louis Caccamo, 1996, p. 51). Cette base commune et homogène permet de traiter l'ensemble des informations qui ne se différencient plus sur un plan analogique mais uniquement en termes de complexité numérique. Elle est un gisement potentiel d'économies d'échelle, qui peut permettre de prolonger la formidable baisse des prix des composants matériels des technologies de l'information et de la communication.

ii -La baisse des prix

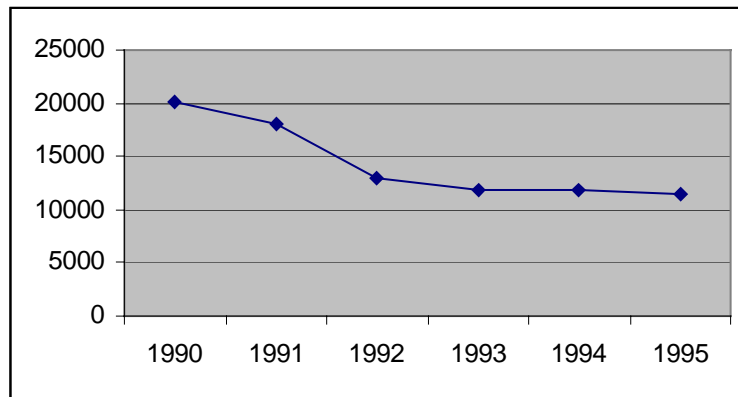
La baisse des prix dans les technologies de l'information et de la communication est particulièrement spectaculaire et sans précédent historique de par son ampleur et sa durée. Sergio Mariotti évalue la baisse du coût d'un ordinateur à 19 % par an sur la période 1958-1992 qu'il compare aux technologies révolutionnaires précédentes : le prix d'utilisation du télégraphe a baissé de 50 % en un siècle (ce qui représente une baisse de 0,7 % par an), le prix

³ Il est prématuré d'évaluer les effets de la fusion AOL Time Warner survenue récemment.

de l'énergie pour l'éclairage de 85 % sur la période 1883-1920, soit 5 % par an (1997, p. 159). Michel Catinat estime que le prix de traitement et de stockage d'une unité d'information a été approximativement divisé par cent tous les dix ans, tout en précisant que la mesure agrégée et précise du phénomène est délicate, tant il est difficile de séparer les effets qualités des effets prix pour les équipements et services liés aux technologies de l'information (1998, p. 38-39).

Par exemple, le prix moyen d'un PC en France a vu son prix moyen "seulement" divisé par deux en cinq ans (20 161 F en 1990 à 11 377 F en 1995) mais sa puissance et ses fonctionnalités se sont considérablement accrus sur la même période.

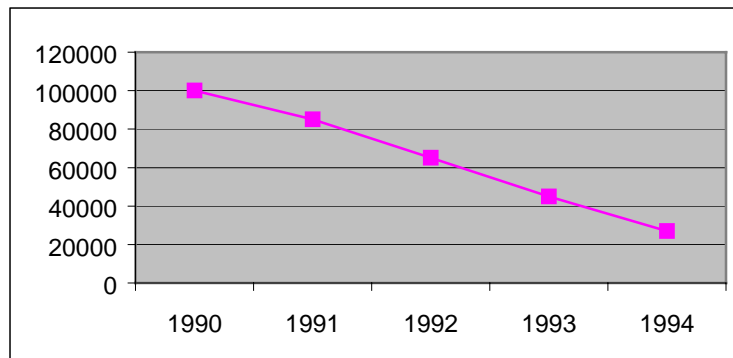
Graphique II
Evolution du prix moyen d'un PC en France (en francs courants)



Source : données IDC, in Eurostaf (1996 A, p. 81).

Pour tenir compte de l'augmentation de la puissance on peut étudier l'évolution du prix du MIPS (Millions d'Instructions par Seconde). On constate que pour les grands systèmes (*mainframes*), le prix du MIPS a été divisé par presque quatre en quatre ans.

Graphique III
Evolution du prix du MIPS grand système (en dollars courants)



Source : données IDC, in Eurostaf (1996 A, p. 78).

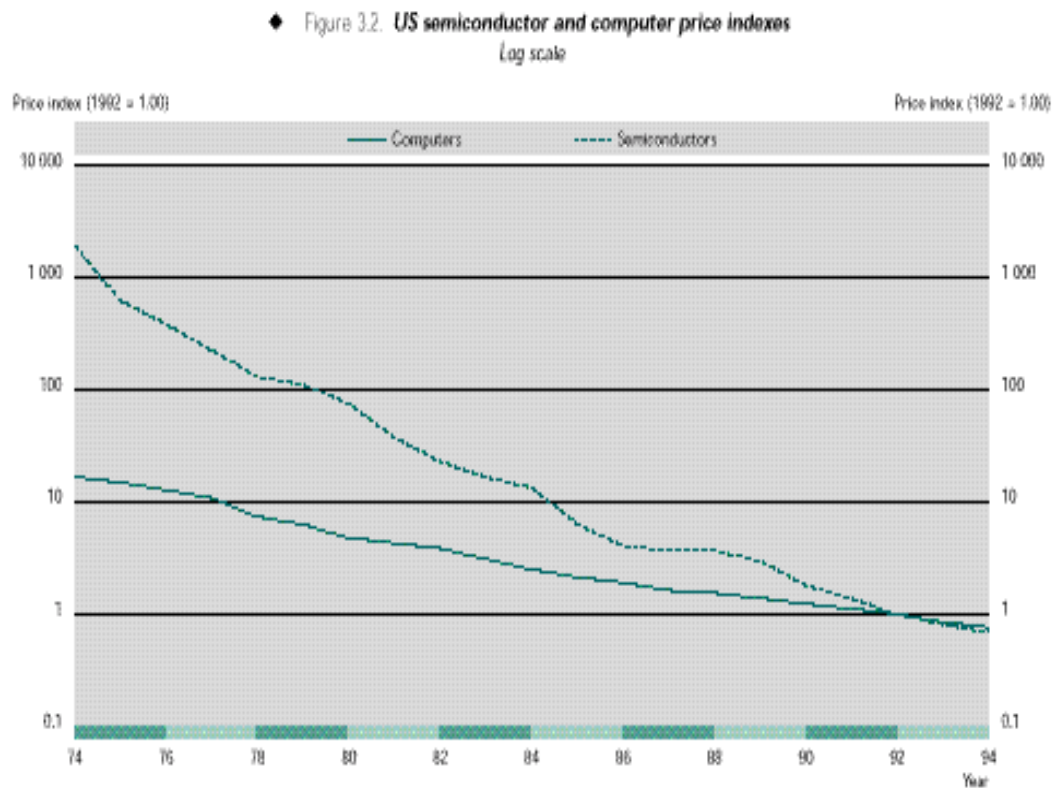
En prenant en compte l'évolution technologique des micro-ordinateurs (effet-qualité), qui ne se limite pas à l'augmentation de leur puissance de traitement, l'INSEE estime que l'indice des prix de vente industriels des micro-ordinateurs a été divisé par dix entre 1988 et 1996 (Secrétariat d'Etat à l'Industrie, 1999, p. 14).

Les estimations peuvent varier significativement selon les méthodes utilisées pour tenir compte de l'effet qualité. La méthode la plus simple et la plus utilisée est le traitement par chaînage (*linking* ou *splicing*) ou "remplacement en dissemblance" qui consiste à "chaîner" les variations de prix du nouveau et de l'ancien produit, en considérant que la différence de prix entre les deux produits à une période t est exactement égale à la différence qualité. Une autre méthode, statistiquement plus convaincante mais plus coûteuse, repose sur une estimation explicite de la différence de qualité (coût des options, estimations du coût de l'amélioration par les producteurs, par des méthodes économétriques souvent qualifiées d'"hédoniques"). Pour le marché des micro-ordinateurs en France, la différence entre ces deux méthodes a été estimée à $-4,4\%$ par an entre 88 et 91 : base 100 au premier trimestre 1988, l'indice des prix des micro-ordinateurs était à 70,9 au premier trimestre 1991 par la méthode de chaînage et à

61,6 en tenant compte de la performance des nouveaux produits (François Lequiller, 1997, p.13-14).

En utilisant des indices des prix hédonistes, on constate que le facteur principal de la baisse des prix des ordinateurs est la baisse du prix des semi-conducteurs (microprocesseurs et mémoires). L'OCDE (1997 B, p. 8) estime que de 1974 à 1994, le prix des semi-conducteurs a été divisé par plus de 1000, celui des ordinateurs par 10.

Graphique III



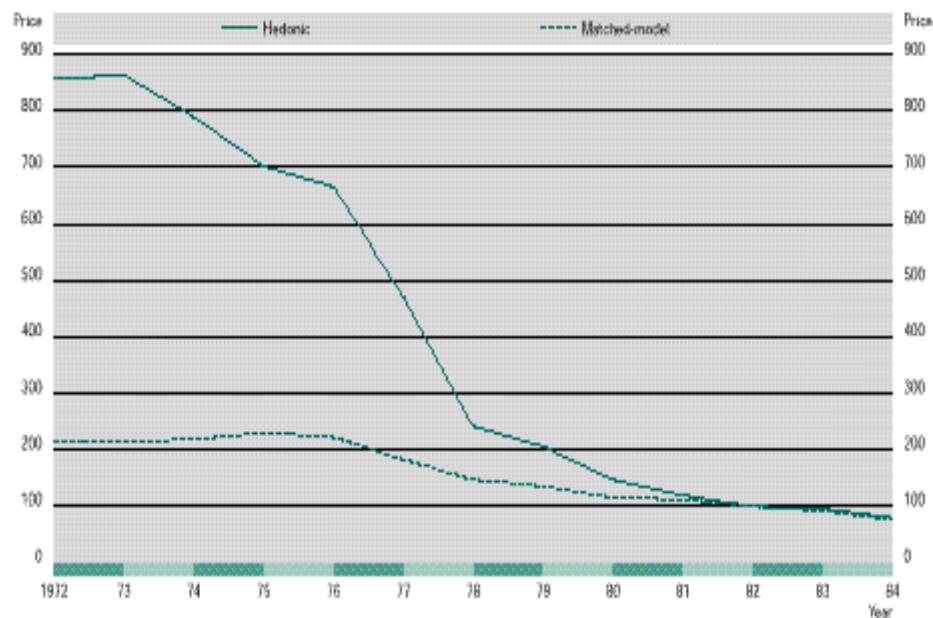
Source: Triplett (1996).

Source : OCDE, 1997 B, p. 8.

Pour les semi-conducteurs également le choix de la méthode utilisée pour mesurer l'effet-qualité a des conséquences importantes (cf. graphique). La différence est spectaculaire entre l'indice des prix basé sur la technique de l'appariement de modèles (prix divisé par deux sur la période) et l'indice des prix basé sur la technique hédoniste (prix divisé par plus de huit).

Graphique IV

◆ Figure 3.1 Comparison of matched-model and hedonic price indexes for US computer processors: 1972-84
1982 = 100



Source: Cole et al. (1986).

Source : OCDE, 1997 B, p. 81.

Le marché des semi-conducteurs présente deux particularités. D'une part, il existe de très fortes instabilités conjoncturelles des prix. Par exemple, en raison de surcapacités le prix des DRAM (mémoires vives dynamiques) sur le marché international est passé en six mois en 1996 de 46-52 dollars à 16-18 dollars (OCDE, 1997 B, p. 69). En sens inverse, le prix des mémoires SDRAM (Synchronous Dynamic Random Access Memory) cadencées à 100 MHz, qui était d'environ 4 dollars l'unité en juillet 1999, était monté à 15 dollars en septembre en raison d'une forte demande sur le marché des ordinateurs (rentrée scolaire, approche de la saison des Fêtes), et l'annonce du séisme à Taïwan a fait instantanément grimper le prix à 20

dollars (selon Dataquest in PC World). Mais, indépendamment de ces fluctuations conjoncturelles, la caractéristique la plus importante des semi-conducteurs est la formidable baisse de leurs "coûts historiques", définis comme étant les coûts dont l'évolution "résulte du progrès technologique introduit par les entreprises d'une industrie particulière" (Jean-Luc Gaffard, 1990, p. 17-18).

Avec la numérisation, cette baisse a contribué à accélérer la réduction du coût des communications : Sergio Mariotti estime que les coûts unitaires de communication ont baissé en proportion de 1000 à 0,01 entre 70 et 90 (1997, p. 160), l'OCDE évalue la baisse du coût d'investissement par voie téléphonique sur liaisons intercontinentales de 6000 dollars en 1989 à 1000 dollars en 1996 (OCDE, 1997 A, p. 11).

iii - un cercle vertueux

La forte baisse des prix des technologies de l'information et de la communication s'insère dans un cercle vertueux "classique" de diffusion, constitué de croissance des volumes, d'économies d'échelle, d'effets d'expérience, d'innovations techniques et de baisse des prix (Eric Brousseau, 1993, p. 175). Ce qui dans le cas des technologies de l'information et de la communication fait que ce processus de diffusion s'est avéré particulièrement spectaculaire, c'est la conjonction de deux phénomènes. D'une part, la baisse des prix élargit continuellement le champ des utilisateurs de ces technologies. D'autre part, la baisse des prix n'entraîne pas une diminution des dépenses des utilisateurs déjà équipés, mais une augmentation des dépenses en matériels plus puissants, en logiciels plus performants, en croissance de trafic, avec la multiplication des fonctionnalités de ces technologies et le développement permanent de nouveaux usages (traitement et transmissions d'images par exemple) (Alain Rallet, 1997, p. 86). La vitesse de propagation de ces technologies semble même s'accélérer, comme l'illustre le processus d'adoption d'Internet, pour lequel on peut parler d'"effet d'avalanche" (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 72).

c - Des technologies "invasives"

On peut considérer les technologies de l'information et de la communication comme une méta-technologie, dont la spécificité par rapport aux grandes innovations précédente serait la généralité de ses applications (Herbert Simon, 1986). Premières technologies "globales" (Sergio Mariotti, 1997, p. 160) ou "polyvalentes" (OCDE, 1997 B, p. 76), elles se diffusent dans tous les secteurs économiques, toutes les activités humaines, toutes les structures sociales. Pouvant intervenir à tous les niveaux des processus productifs, elles concernent

l'ensemble des travailleurs (industrie et services, ouvriers, employés et cadres) et leurs possibilités apparaissent sans limite si ce n'est celle de l'imagination, même s'il existe des incertitudes importantes sur leur concrétisation, en raison de l'ampleur et de la nature des changements nécessaires (Michel Catinat, 1998, p. 41-42).

Le caractère diffus et envahissant des technologies de l'information et de la communication est particulièrement marqué dans le cas de l'informatique, phénomène qu'IBM a baptisé "*pervasive computing*". "L'informatique est par nature une technologie invasive susceptible d'être appliquée à d'innombrables usages par des utilisateurs très divers" (Gérard Dréan, 1996 A, p.311). Cette infinie variété d'usages a pour conséquence que la part relative de matériels et de services informatiques au sens strict (c'est à dire à caractère universel), risque de décroître au profit d'objets informatisés très divers et spécialisés pour une utilisation particulière (se substituant à des objets non informatisés) et de services informatisés (correspondant à l'informatisation d'activités déjà existantes). Cette évolution, avec la présence de microprocesseurs pour le traitement de l'information dans l'ensemble des biens, peut être comparée sur une échelle plus importante à celle qui s'est produite avec la diffusion des moteurs électriques. En faisant migrer des activités qui appartenaient initialement à l'industrie informatique vers les industries et les services utilisateurs, Gérard Dréan estime qu'à terme "l'industrie informatique perdra ses frontières et par-là même l'essentiel de son individualité" et que "l'histoire de l'informatique serait alors terminée en tant que telle" (1996 A, p. 355).

d - Les effets potentiels des technologies de l'information et de la communication

On peut décrire les possibilités qu'offrent objectivement le développement des technologies de l'information et de la communication, qu'illustrent des exemples concrets d'applications déjà réalisées, tout en soulignant que leur généralisation et leurs modalités (plurielles) d'évolution dépendent également de facteurs non technologiques (cf. 3). Ces possibilités sont jugées exceptionnelles, sans comparaison avec les progrès technologiques observés jusqu'à ce jour, à l'exception peut-être de l'imprimerie et de l'utilisation de la vapeur ou de l'électricité (Michel Catinat, 1998, p. 37).

Elles concernent tout d'abord leurs facultés à accélérer le temps et à agrandir l'espace, avec des conséquences importantes sur les espaces de temps et de lieu des activités économiques (Pascal Petit, 1998, p. 18). Les développements récents en matière d'information

privilégient trois axes : "l'ubiquité" avec la possibilité d'accéder à la même information sur n'importe quelle machine (ordinateur, téléphone, TV, console...), la "personnalisation" avec l'adaptation de l'information aux besoins de l'utilisateur et "l'instantanéité" avec un accès en temps réel à une information mise à jour en permanence⁴.

Les technologies de l'information et de la communication favorisent le développement des connaissances codifiées. Certes, les perfectionnements des technologies de stockage, d'enregistrement, de recherche et de diffusion des messages n'affectent pas directement la codification des connaissances, qui dépend du développement de nouveaux langages, de l'évolution des aptitudes à modéliser les phénomènes et les activités, et des avancées des technologies de codification (Robin Cowan, Dominique Foray, 1998, p. 311). Mais ils ont une influence importante indirecte sur l'augmentation du stock de connaissances codifiées pour deux raisons liées. D'une part, la baisse du coût du stockage et de la diffusion des connaissances codifiées, et l'élargissement du nombre d'utilisateurs augmentent le bénéfice social net qui résulte de la codification. D'autre part, il devient intéressant de codifier des connaissances de plus en plus complexes, qui requièrent de larges quantités d'informations variées (idem, p. 312-313).

De ce fait, "les progrès des technologies de l'information et de la communication permettent désormais de traiter, de stocker et de transmettre sans contrainte de temps ni d'espace ainsi que dans un rapport interactif, le savoir, le savoir-faire et le faire-savoir. En d'autres mots, le rapport social à distance a les traits de la communication vive à l'exception du toucher et de l'odorat. Au monde physique se superpose un monde du rapport social "en temps réel" sans contraintes spatiales" (A. Stekke, 1996, p. 1). "La programmabilité des technologies de l'information et de la communication permet d'incorporer à un support physique, reproductible et appropriable, une ressource auparavant difficilement transférable : le savoir-faire" (Eric Brousseau, 1993, p. 207). Avec la création de supports de l'information, qui ne sont plus simplement des supports de stockage mais qui ont la capacité de traiter l'information et qui sont accessibles à distance, les compétences et les connaissances nécessaires pour accomplir son travail, sont de plus en plus intégrées à des artefacts, ce qui permet de les faire circuler indépendamment des êtres humains et de les transformer en

⁴ Ces évolutions ont conduit à la constitution d'un label U.P.I. pour Ubiquité, Personnalisation, Instantanéité (La Lettre de la NetEconomie, n° 14, 8/10/1999)

marchandises informationnelles (Abbe Moshowitz, 1997, p. 23). La "matérialisation" de la compétence dans un système technique permet de dupliquer et de vendre la compétence, qui n'est plus totalement tributaire de son stockage par les individus et de sa difficulté de transfert (Eric Brousseau, 1993, p. 207). Les supports de l'ère pré-informatique (comme les livres) étaient des auxiliaires de la mémoire, qui pouvaient assister les êtres humains dans la production des connaissances, mais ne pouvaient par eux-mêmes ni créer, ni participer directement à l'application des connaissances, ce qui restait l'apanage des êtres humains. La reproduction des connaissances d'une société dépendait donc entièrement de l'acquisition des connaissances par chaque génération successive, alors que les systèmes informatiques savent générer de nouvelles connaissances et utiliser des connaissances existantes pour assumer des fonctions indépendamment de l'être humain (Abbe Moshowitz, 1997, p. 27). Jean-Daniel Dessimoz parle de "systèmes cognitifs artificiels" (ACS) pour désigner cette possibilité de comportement réactif complexe, ce qui est qualitativement nouveau pour des machines faites par l'homme (1997, p. 230). Les avantages économiques de ces marchandises informationnelles (faible coût de leur reproduction, malléabilité, adaptabilité) font qu'elles sont promises à un rapide développement, ce qui peut être considéré "comme une extension de la marchandisation du travail engendrée par l'industrialisation" (Abbe Moshowitz, 1997, p. 33). Cette évolution, qui "touche aux relations humaines les plus fondamentales (...) peut s'avérer encore plus profonde que les changements apportés par l'industrialisation" (idem).

Les technologies de l'information et de la communication peuvent avoir également d'importantes conséquences sur la variété des produits (biens et services). Elles ouvrent des potentialités considérables de création de nouveaux types de services et de nouvelles prestations (Faïz Gallouj, Olivier Weinstein, 1997)⁵. Elles permettent de commercialiser des services jusqu'alors non négociables (Sergio Mariotti, 1997, p. 160). En permettant l'intégration de données très diverses, elles offrent de nouvelles possibilités de création culturelle. Elles permettent d'intégrer dans la vente des biens une part croissante de services, et de diversifier considérablement les produits fournis ; par exemple, le nombre moyen de produits vestimentaires offerts par les fabricants américains a augmenté de 63 % entre 1988 et 1992, avec également une augmentation importante du nombre de nouveaux produits et des produits abandonnés, ce que l'on peut mettre en relation avec le fait que, sur la même période;

⁵ Armand Hatchuel (1996) estime que 30 à 40 % des produits sur les marchés n'existaient pas il y a cinq ans.

la proportion d'entreprises de vêtements qui utilisaient l'échange électronique des données (EDI) est passé de moins de 5 % à plus de 30 % (OCDE, 1997 B, p. 204). Les technologies de l'information et de la communication peuvent permettre aux entreprises de gagner en flexibilité et en réactivité par rapport à des marchés plus variables.

C'est également une conséquence des changements permis par les technologies de l'information et de la communication à tous les niveaux des processus productifs (Pascal Petit, 1998, p. 18). Les technologies de l'information et de la communication prolongent un mouvement de rationalisation entamé depuis le XIX^{ème} siècle, en améliorant le contrôle au sens cybernétique du terme (systématisation de la collecte d'information) et la rationalisation des décisions (systématiser l'optimisation et l'effort de formalisation qui conduit à concevoir des procédures plus efficaces). Elles étendent l'automatisation de la production directe à la sphère de l'encadrement de la production et du travail indirect (Jean-Louis Cacomo, 1996, p. 15). Un nombre sans cesse accru d'opérations de manipulations de symboles, dont l'importance est croissante par rapport aux opérations manuelles, sont exécutées par des machines avec une grande rapidité, fiabilité et à faible coût (Eric Brousseau, 1993, p. 214). Il devient dès lors possible d'expérimenter des procédures différentes à ce niveau, vu que les machines sont plus faciles à manipuler que les hommes (Abbe Moshowitz, 1997, p.31). Par exemple, dans la fabrication des vêtements aux Etats-Unis, le doublement des investissements annuels consacrés aux systèmes d'information pour obtenir une gestion des stocks à "réaction rapide" a permis de réduire les stocks de 60 % depuis 1985 (OCDE, 1997 B, p. 17). On peut également citer l'exemple de l'entreprise Wal-Mart qui est devenu le premier détaillant mondial grâce notamment à des investissements très élevés⁶ dans un système de distribution informatisé, qui lui permet de renouveler automatiquement ses stocks mais également de faire de la modélisation prédictive (OCDE, 1997 B, p. 207). Un autre exemple est dans le domaine de la conception, où les technologies de l'information et de la communication ont permis à 4 000 ingénieurs et techniciens de Boeing, répartis en 230 équipes de conception interfonctions, de travailler simultanément à la conception du Boeing 777. Les gains estimés sont l'économie de plus d'un million d'heures de travail, l'élimination de 50 % des défauts de

⁶ Les investissements de Wal-Mart dans les technologies de l'information et de la communication ont atteint 600 millions de dollars entre 1987 et 1992. Ils ont permis à Wal-Mart de réaliser des économies estimées à un milliard de dollars par an par rapport à ses concurrents, en ayant des coûts de distribution beaucoup plus faibles, ce qui s'est traduit par un taux de croissance annuel moyen pour Wal-Mart de 30,6 % entre 1983 et 1993 (OCDE, 1997 B, p. 207).

conception, la réduction du temps de développement du nouvel appareil de 25 % et l'approbation de l'organisme de réglementation en un temps record (OCDE, 1997 B, p. 208). Enfin, une des potentialités importantes des technologies de l'information et de la communication réside dans leurs capacités à intégrer les processus de conception, de production et de distribution des biens et services ce qui leur "offrent de nouveaux moyens pour réduire les coûts, améliorer la qualité des produits, faciliter un contrôle flexible, favoriser l'adaptation aux évolutions du marché, et renforcer l'innovation organisationnelle et technique" (Abbe Moshowitz, 1997, p.30). La forme la plus avancée de cette évolution est la réintroduction de la production "sur mesure" mais à bas prix, dont l'exemple le plus célèbre est l'entreprise Levi's qui produit des jeans personnalisés à partir de la saisie des mesures et des souhaits du client sur Internet.

Si les technologies de l'information et de la communication sont des "technologies de production", ce sont également des "technologies de transaction (...) mises en œuvre pour réaliser une partie des tâches de coordination induites par la division du travail : échange marchand, coopération, régulation hiérarchique, etc." (Eric Brousseau, 1993, p. 20). Elles permettent potentiellement d'améliorer les "capacités organisationnelles des entreprises" (Alfred D. Chandler, 1992) par une efficacité accrue du pilotage, une meilleure allocation des ressources, la réduction des coûts de coordination, et une plus grande efficacité des mécanismes incitatifs (individualisation des salaires par exemple) (Eric Brousseau, 1993, p. 228-236). Sont le plus souvent évoqués sur le plan interne, l'aplatissement des structures hiérarchiques et la responsabilisation des unités (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 42) et la montée en puissance des équipes d'experts en nouvelles technologies (Faridah Djellal, 1995, p. 39). Sur le plan externe, la diffusion des relations réticulaires (association/coopération au sein de réseaux réels ou virtuels) contribue à une certaine dilution des frontières de la firme (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 62). Enfin, la réduction des coûts de transaction devrait logiquement favoriser le "faire faire" au détriment du "faire", par l'externalisation de certaines fonctions, le développement de la sous-traitance.

Les technologies de l'information et de la communication peuvent avoir également un impact sur la société globale. Elles contribuent à transformer les grandes infrastructures de la société civile, en agissant directement pour les communications, et indirectement par substitution/complémentarité pour le transport et l'énergie (Sergio Mariotti, 1997, p. 162). Elles ont accompagné les phénomènes de globalisation et de mondialisation des marchés.

Elles ont participé à la transformation des marchés financiers en des marchés "quasi-parfaits", avec leurs conséquences sur les relations entre capital industriel et capital financier.

Selon certains auteurs, le "paradoxe de Solow" appartient à une époque révolue⁷, et les technologies de l'information et de la communication sont devenues un facteur endogène de croissance (Michel Catinat, 1998, p. 40). Aux Etats-Unis, un récent rapport du Department of Commerce estime que le secteur des technologies de l'information qui représente 8 % du PIB américain, a contribué à hauteur de 35 % à la croissance totale de l'économie entre 1995 et 1998 et a créé plus de 5 millions d'emplois durant cette période. Une des explications fréquemment avancées du différentiel de croissance entre les Etats-Unis et l'Europe sur la période récente, réside dans la différence adoptée par rapport à l'adoption de ces technologies : en 1996 les entreprises européennes n'ont investi dans les technologies de l'information que 936 dollars par employé contre 1598 dollars aux Etats-Unis ; elles ne consacrent que 4,3 % de leur chiffre d'affaires à l'informatique à comparer aux 6,2 % des entreprises américaines (Eurostaf, 1997 A, p. 5-6)⁸.

2 - La base d'un nouveau paradigme ?

En s'appuyant sur les potentialités des technologies de l'information et de la communication à transformer profondément les processus de production et de distribution que Alfred D. Chandler compare aux effets du chemin de fer, du télégraphe, du bateau à vapeur et du câble, dans le domaine des réseaux de transports et de communications pour la révolution industrielle (1992, p. 56), certains auteurs concluent à l'avènement d'un nouveau paradigme qualifié de technologique, de technico-économique ou de socio-économique.

a - Les paradigmes technologiques

Le concept de paradigme a été introduit en sociologie de la recherche fondamentale par T.H. Kuhn (1970). A partir d'une similitude, en termes de définition et de procédure, avec les paradigmes scientifiques, Giovanni Dosi définit des paradigmes technologiques. Selon

⁷ Ce constat confirmerait une explication couramment avancée du paradoxe de Solow, à savoir que sa validité provisoire s'expliquait par la lenteur des processus d'apprentissage et les délais historiques d'ajustement aux nouvelles technologies (cf. chapitre III).

⁸ De même l'OCDE montre qu'il existe une forte corrélation ($R^2=0,69$) pour les différents pays de l'OCDE entre les dépenses de technologies de l'information par habitant, et le PIB par habitant (OCDE, 1993 B, p. 19). Il faut toutefois préciser que des liens de causalité entre ces deux phénomènes peuvent exister dans les deux sens.

Giovanni Dosi, la technologie est "un ensemble de connaissances à la fois directement "pratiques" (liées à des problèmes et des appareils concrets) et "théoriques" (mais applicables en pratique bien que non nécessairement déjà mis en pratique), de savoir-faire, de méthodes, de procédures, d'expériences de succès et d'échec et aussi, bien sûr, d'instruments et d'équipements physiques" (Giovanni Dosi, 1982, p. 151-152). Un paradigme technologique est "un modèle de résolution de certains problèmes technico-économiques fondé sur des principes de connaissance hautement sélectionnés dérivés des sciences, conjointement avec des règles spécifiques conçues pour acquérir de nouvelles connaissances et les préserver, dans la mesure du possible, contre une rapide diffusion vers les concurrents" (Giovanni Dosi, 1988, p. 1127). Un paradigme technologique est une façon de penser l'état donné des solutions techniques et sa logique d'évolution, une vision des solutions acquises temporairement et des problèmes posés, des directions et des modalités du changement technique, à partir d'un programme de recherche. Par exemple, le paradigme électronique s'est constitué autour de l'agrandissement permanent d'une "base de connaissances", dont le noyau était les innovations radicales que constituaient le transistor, puis les circuits intégrés (Giovanni Dosi, 1984).

b - Les paradigmes technico-économiques

En étudiant le changement technologique sous un angle plus macroéconomique, Chris Freeman parle de paradigme technico-économique, "l'expression technico-économique [soulignant], plus que paradigme technologique, le fait que les changements impliqués dépassent le cadre technologique spécifique d'un produit ou d'un processus et affectent la structure des coûts d'entrée, ainsi que les conditions de production et de distribution à travers l'ensemble du système" (Chris Freeman, 1988, p. 48). Chris Freeman prend en compte "l'interdépendance existant entre certains groupes d'innovations du secteur technique, du management et de l'organisation, ainsi que le rôle du cadre institutionnel et social dans lequel se poursuit la diffusion" (idem, p. 46). Il établit une taxonomie du changement technologique en quatre catégories : les innovations progressives, plus ou moins permanentes et dont l'unique objet est le perfectionnement de la gamme des produits et des procédés existants ; les innovations radicales, événements discontinus introduisant des ruptures, dont la traduction est l'adjonction de nouvelles lignes et de nouvelles colonnes dans la matrice des tableaux entrées-sorties ; les systèmes de nouvelles technologies, "constellations" d'innovations entre lesquelles existent des interrelations techniques et économiques ; les changements de paradigme technico-économique qui entraînent de nombreux groupes d'innovations radicales et

progressives et renferment parfois un certain nombre de systèmes de nouvelles technologies (idem, p. 47).

La spécificité d'un changement de paradigmes technico-économiques réside dans l'impact qu'il exerce sur l'ensemble de l'économie, en affectant directement ou indirectement la presque totalité des branches de l'économie. Cette conception correspond à la "force de destruction créatrice" de la théorie des cycles longs de Schumpeter. Un nouveau paradigme technico-économique, "orientation technologique prédominante pour une phase entière de croissance et de développement économiques" (idem, p. 49), se développe initialement à partir de l'ancien paradigme, révélant ses avantages déterminants sur le plan technique, économique et social pendant la phase de dépression du cycle de Kondratiev précédent. A la base de tout paradigme nouveau, il existe un input ou un ensemble particulier d'inputs, "facteur clé" de ce paradigme, qui doit posséder des coûts très modérés et en baisse rapide, des possibilités d'approvisionnement pratiquement illimitées à long terme, et un potentiel manifeste d'utilisation quasi universelle (C. Perez, 1983). Ce facteur clé (ou "technologie permissive" selon l'expression de R. Barras) a été successivement le charbon et les transports à vapeur, l'acier, puis le pétrole. Les technologies de l'information et de la communication, technologies à très grande diffusion, porteuses d'une multitude d'applications et de possibilités nouvelles intéressant l'ensemble du système économique, présentent aujourd'hui les caractéristiques nécessaires à l'avènement d'un nouveau paradigme technico-économique (Chris Freeman, 1995, p. 35), et pourraient être le moteur d'un cinquième Kondratiev. Au-delà de ces dimensions technologiques (rôle déterminant de l'information à la place de l'énergie), ce nouveau paradigme intègre de multiples changements concernant les produits (différenciation, particularisation et évolution rapide des biens et des services), les équipements (systèmes de production flexibles), la place relative des services (les produits servent de support aux services, qui pour la première fois connaissent des changements techniques rapides), l'organisation de la production (recomposition du processus productif et intégration du design, du management, de la production et du marketing), l'organisation des entreprises (structures horizontales, intelligence répartie), les relations entre entreprises (réseaux d'entreprises), l'organisation du travail (main d'œuvre polyvalente)...

Cette conception des paradigmes technico-économiques comprend donc d'importantes dimensions organisationnelles et institutionnelles, complémentaires des changements technologiques. Toutefois les modifications sociales et institutionnelles apparaissent subordonnées aux évolutions technologiques : "les nouvelles technologies exigent une

transformation des techniques de gestion et des modes d'organisation ; très souvent elles exigent aussi de nouvelles formes de réglementation du marché et de nouveaux types de normes nationales et internationales" (Chris Freeman, 1995, p. 35). De même, C. Perez (1983) estime que les périodes de croissance soutenue sont des périodes où le cadre institutionnel est en phase avec le paradigme technico-économique. La crise survient quand l'émergence d'un nouveau paradigme technico-économique (due à l'irruption des innovations majeures, techniques et organisationnelles) ne peut plus être couplée avec les institutions existantes. Ainsi, selon Chris Freeman la situation actuelle se caractérise par un "processus d'ajustement long et douloureux" qui n'a "cependant pas encore amené tous les changements institutionnels et sociaux nécessaires pour favoriser la diffusion du nouveau paradigme technico-économique" (1995, p. 28), "l'immense potentiel de productivité que recèlent les technologies de l'information et de la communication [étant] neutralisé par les rigidités du système social" (idem, p. 32).

c - Les paradigmes socio-économiques

La conception de Chris Freeman a suscité des critiques. Michel Callon souligne que l'analyse de la genèse des innovations technologiques doit intégrer les aspects économiques, politiques et sociaux qui se manifestent dans les conflits entre de multiples acteurs dans la création scientifico-technique. Si ces facteurs interviennent pour Chris Freeman dans l'analyse des innovations secondaires ou d'accompagnement, ce n'est pas le cas pour les innovations majeures qu'il considère comme étant uniquement le produit de la logique propre du développement des connaissances scientifiques de base (Michel Callon, 1989, p. 71). De même, Pierre Dockès considère que "les aspects conflictuels à la source des innovations plurielles, les modalités spécifiques (...) de régulation des conflits et des innovations ne sont pas pris en compte" (1990, p. 39). De façon plus générale, Pierre Dockès souligne que dans la conception des paradigmes technico-économiques, "si la dimension sociale est présente, c'est "à côté" et non au sein du paradigme productif" et qu'il existe "une dichotomie entre, d'une part, le dynamisme innovateur au sein du paradigme technico-économique et, d'autre part, le contexte socio-économique qui doit s'adapter pour que la mutation globale se fasse et devienne performante" (idem).

Avec l'objectif d'éviter tout "déterminisme technologique", d'articuler étroitement les trois dimensions socio-organisationnelle, technique et économique, solidaires et inséparables, Pierre Dockès introduit le concept de paradigme socio-économique. Celui-ci ne peut être

appréhendé que dans sa globalité, il ne se structure plus autour d'un facteur clé, mais d'un rapport de production ou d'une forme particulière de celui-ci. Un paradigme socio-économique est "une façon dominante de penser la production au sens large (pas seulement le procès de travail), c'est à dire l'organisation sociale, économique et technique de la production partagée par l'ensemble des entrepreneurs et "décideurs" et qui tend à être diffusée dans l'ensemble de la population concernée" (Pierre Dockès, 1990, p. 39-40). A partir des mêmes préoccupations d'une approche plus sociale et institutionnelle des paradigmes, Faridah Djellal (1995) parle de paradigmes socio-techniques pour insister sur l'importance des interactions entre technologie et innovations organisationnelles et sociales, la technologie n'étant plus considérée comme le seul moteur de l'innovation sociale.

Cette volonté de combiner les aspects techniques, économiques, sociaux, culturels dans leurs multiples interrelations et sans hiérarchisation a priori, nous semble particulièrement intéressante et s'inscrire dans la même démarche que celle que nous avons utilisé pour analyser l'évolution de l'informatique comme la constitution successive de réseaux technico-économiques (cf. chapitre II). Par contre le terme de paradigme, appliqué à un cadre d'analyse macroéconomique et/ou macrosocial, nous semble problématique dans la mesure où il induit la vision d'un seul paradigme qui s'impose massivement où "on ne peut pratiquement plus penser autrement la production" (Pierre Dockès, 1990, p. 42). Certes, les différents auteurs envisagent l'existence de plusieurs paradigmes, mais les paradigmes "alternatifs" restent "marginaux", voire "purement virtuels" (idem p. 41), et c'est seulement pendant les périodes de transition entre deux paradigmes dominants, que coexistent momentanément l'ancien et le nouveau paradigme avant que ce dernier ne s'impose (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 57). Il nous semble au contraire qu'une vision "d'abord sociale et culturelle" et non "pas principalement technico-économique" des "paradigmes" (Pierre Dockès, 1990, p. 44) peut permettre de comprendre l'existence durable de plusieurs "paradigmes" différents, ce qui ne signifie pas qu'ils soient immuables.

3 - La coexistence permanente de plusieurs "paradigmes" différents

Les trajectoires technologiques ne sont pas seulement technologiques

Certes il existe des trajectoires technologiques, au sens où dans l'arbre des orientations possibles, les choix effectués à chaque étape éliminent d'autres directions qui étaient possibles (Pierre Dockès, 1990, p. 33-34). Ces trajectoires technologiques se caractérisent par la présence d'irréversibilités, de rendements croissants d'adoption, l'importance des "petits

événements" initiaux qui impliquent l'existence d'une contrainte de sentier. La numérisation est un exemple d'une telle trajectoire dans le traitement du signal, avec les bifurcations essentielles qu'ont constituées le choix du numérique par rapport à l'analogique, et, au sein du numérique, le choix de la représentation binaire. Si ces trajectoires sont technologiques au sens où elles concernent des choix techniques, les déterminants de ces choix eux sont multiples. Par exemple, dans le succès du numérique binaire, les considérations économiques (possibilité de produire des composants peu coûteux qui permettait d'introduire la logique de Boole) ont été essentielles (Jean-Louis Caccamo, 1996, p. 43).

De façon plus générale beaucoup d'évolutions qui apparaissent comme d'essence essentiellement technologique ne peuvent être comprises sans intégrer les dimensions économiques, sociales voire culturelles. Il en est ainsi de la "loi" de Gordon Moore (cofondateur d'Intel) énoncée fin des années soixante, selon laquelle la puissance des microprocesseurs et la capacité des mémoires doublent tous les 18 mois (cf. chapitre III). Cette "loi", qui s'est remarquablement vérifiée jusqu'à présent, est souvent considérée comme ayant "acquis une dynamique largement autonome, d'une régularité et d'une ampleur impressionnante" (Pascal Petit, 1998, p. 354). Or, si la vérification de cette loi nécessitait certes l'existence de possibilités technologiques (en termes de miniaturisation), celles-ci n'ont pu se réaliser qu'au prix d'une croissance spectaculaire des investissements. Pour les mémoires, Siemens fournit les chiffres suivant :

Tableau XII
Coûts fixes de production des mémoires (en millions de dollars courants)

Capacité	Introduction	R. & D.	Production
256 Kb	1985	400	180
1 Mb	1988	350	1300
4 Mb	1990	380	1600
16 Mb	1993	510	3700
64 Mb	1996	1000	6000

Source : Siemens, cité par Gérard Dréan, 1996 A, p. 126

Il est évident que sans les changements économiques et sociaux qui ont contribué à la diffusion massive de tels composants électroniques, de tels investissements n'auraient pu être rentabilisés et que la "loi" ne se serait pas appliquée.

De même, la loi de Grosh, dont l'inversion dans les années quatre-vingt explique le formidable succès de la microinformatique, est souvent interprétée comme étant d'essence technologique. La loi de Grosh affirmait qu'à coût égal un grand système informatique offrait plus de puissance que plusieurs petits systèmes. Cette loi qui était vérifiée dans les années soixante, va s'inverser radicalement (cf. tableau XIII), cette inversion s'amplifiant considérablement dans la période récente (cf. tableau XIV).

Tableau XIII
Inversion de la loi de Grosh
MIPS (millions d'instructions par seconde) pour un million de dollars courants

Ordinateur	1950	1960	1970	1980	1984
Grands systèmes (IBM 3084 en 1984)	0,03	0,11	0,6	1,2	2,1
Petits systèmes (IBM PC en 1984)	0,02	0,05	0,5	14,0	100,0

Source : Philippe Breton, 1987, p. 196

Tableau XIV
Evolution des coûts micros/mainframes

Dollar/MIPS (processeur)	Micros	Mainframes	Ratio
1990	700	100 000	143
1995	60	55 000	917
Dollar/Mega-Octet (mémoire centrale)	Micros	Mainframes	Ratio
1990	90	6 000	67
1995	6	1 500	250

Source : Siemens, cité par Gérard Dréan, 1996 A, p. 50.

Or les technologies utilisées ne sont pas radicalement différentes entre les grands systèmes et les micro-ordinateurs. Les différences constatées s'expliquent bien davantage par des facteurs économiques : la standardisation dans la micro-informatique a favorisé la spécialisation des entreprises sur des segments distincts et l'existence d'économies d'échelles, qui ont généré des cercles vertueux d'augmentation des volumes et de baisse des prix.

Par contre la prise en compte des facteurs socio-économiques n'induit pas qu'une trajectoire technologique puisse être considérée comme "la direction de l'évolution à partir d'un paradigme technologique" (Giovanni Dosi, 1982). L'existence de trajectoires technologiques n'empêche pas la coexistence durable de différents paradigmes, considérés comme des "façons de penser la production". Dans la constitution d'un paradigme interviennent les techniques, mais également les institutions économiques et sociales, les représentations et les stratégies d'acteurs individuels et collectifs, "où chacun cherche à infléchir à son avantage un environnement dont il apprend les règles au fur et à mesure de leur émergence, sans avoir toujours conscience de la contribution en partie non intentionnelle de son action à cette émergence" (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 64). Si l'on rejette l'idée que les institutions et la position relative des groupes sociaux seraient façonnées par les innovations technologiques, et qu'au contraire les institutions sont le produit de l'histoire (Pierre Dockès, 1990, p. 34), on peut comprendre qu'il se produise simultanément et durablement des évolutions différentes dans les façons de produire, que nous détaillerons ultérieurement (cf. chapitre VII). *De façon quelque peu schématique, le développement des technologies de l'information et de la communication peut tout à la fois contribuer à la constitution d'un nouveau paradigme technico-économique où "la diversité et la flexibilité à tous les niveaux remplacent l'homogénéité et la rigidité des systèmes" (Chris Freeman, 1988, p. 53), mais également offrir de nouvelles opportunités pour le développement du "paradigme fordiste".* En effet, les institutions et les pratiques qui influencent les évolutions ne sont pas identiques, elles sont le produit d'histoires différentes, selon les pays, selon les secteurs, voire selon les entreprises : "les points de contacts entre grappes d'innovation technologique et organisationnelle sont multiples, liés au contexte, au secteur, au pays" (Sergio Mariotti, 1997, p. 164).

Des différences selon les pays

Au niveau national, comme le souligne Chris Freeman, "les structures institutionnelles diffèrent considérablement d'un pays à l'autre puisqu'elles sont le reflet de l'histoire propre à

chaque pays, des conflits sociaux et politiques qui l'ont agité, des guerres, de la division internationale du travail, des traditions culturelles et de bien d'autres facteurs encore" (1995, p. 28). C'est ce que montrent les approches en termes de systèmes nationaux d'innovation qui insistent sur les spécificités nationales des schémas d'interrelations au sein des réseaux d'institutions dans les secteurs publics et privés (Jorge Niosi, Bertrand Bellon, Paolo Saviotti, Michaël Crow, 1992). Ces approches peuvent contribuer à expliquer les singularités de l'approche française des technologies de l'information et de la communication, "souvent considérées comme un simple moyen d'automatiser des processus existants, dans une perspective taylorienne élargie jusqu'aux services" (Daniel Kaplan, 1999). Ce constat est partagé par Ghislaine Clot-Lafleur et Hervé Passeron, qui soulignent, qu'en France, les technologies de l'information sont "conçues uniquement par la plupart des entreprises, comme un moyen de rationaliser l'organisation du travail, de "faire de la productivité" en remplaçant les hommes par des machines, ce qui est cohérent avec le fait de se situer trop souvent sur le seul terrain de la compétitivité/prix en négligeant la compétitivité hors prix" (1997, p. 130-131). Cette spécificité française a des racines profondes et multiples comme le montre l'étude de Marc Maurice, François Sellier et Jean-Jacques Silvestre (1982) qui au terme d'une comparaison entre la France et l'Allemagne concluaient que des entreprises technologiquement comparables présentaient des différences dans leurs organisations, leurs systèmes de travail, leurs structures hiérarchiques et d'encadrement.

Des différences inter et intrasectorielles

Le cadre socio-institutionnel pertinent n'est pas uniquement national, il est également sectoriel, ce qui peut expliquer des différences intersectorielles dans les conséquences de l'introduction des technologies de l'information et de la communication. Par exemple, dans le transport routier Ghislaine Clot-Lafleur et Hervé Passeron montrent comment l'introduction des bourses de fret télématiques a entraîné une précarisation de l'emploi (1997, p. 133), contrairement au postulat de "la plus grande stabilité de l'emploi qui devrait résulter de l'utilisation des technologies de l'information et de la communication " (Chris Freeman, 1995, p. 31). Si dans certaines branches d'activité, ces technologies ont permis d'augmenter la variété et la personnalisation des produits, dans d'autres activités (notamment de services, par exemple une partie de la restauration ou de l'hôtellerie) elles ont contribué à une plus grande standardisation des prestations. De même, dans certains secteurs, les technologies de l'information et de la communication peuvent permettre de développer des transactions directes (sans intermédiaires), alors que d'autres secteurs voient l'apparition de nouveaux

intermédiaires⁹, des "métamédiateurs" proposant des prestations intégrées (dont l'archétype est American Express autour de sa carte de paiement) ou au contraire des "inframédiateurs" dont l'originalité est d'avoir une offre particulièrement ciblée et réactive, comme Amazon.Com aux Etats-Unis ou Dégri¹⁰Tour en France (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 45).

Enfin à l'intérieur d'un secteur des différences significatives peuvent également exister bien que la base technologique apparaisse comme assez homogène. C'est notamment ce qu'ont mis en évidence des études sur certains services (tourisme, services d'aide aux personnes âgées). Un des éléments d'explication des différences intrasectorielles constatées est le fait qu'il y a codétermination de la technologie par ses producteurs et ses utilisateurs à travers des schémas complexes d'apprentissage de ses usages et de découverte de ses potentialités (Nathan Rosenberg, 1983). Selon Manuel Castells (1998), un des traits distinctifs de la révolution des technologies de l'information par rapport aux autres révolutions industrielles, est le fait qu'utilisateurs et acteurs se confondent "les utilisateurs [ayant] ainsi la possibilité de devenir maître de la technologie, comme dans le cas d'Internet". Ce rôle accru des utilisateurs dans l'évolution technologique est également pris en compte par Gennady M. Dobrov (1979) à partir de la notion *d'orgware*, définie comme la "la composante structurale d'un système technologique, spécialement conçue pour intégrer l'homme et ses compétences professionnelles et assurer le fonctionnement du *hardware* et du *software* du système ainsi que l'interaction de celui-ci avec d'autres éléments et avec des systèmes de nature différente". L'importance prise par *l'orgware* correspond à la "croissance de la fonction sociale de l'homme dans le système de production", qui explique que les caractéristiques organisationnelles ont un impact déterminant sur les utilisations d'une technologie. La diversité des *orgware* peut permettre d'expliquer les différences constatées dans la "façon de penser la production" à partir de la diffusion des mêmes technologies.

Les interrelations entre technologies et structures organisationnelles

Les choix effectués en termes de conception organisationnelle, produits de l'histoire et des rapports de force au sein de l'organisation, ont une influence importante sur les effets des

⁹ Beaucoup de sociétés qui ont connu récemment des succès spectaculaires sont le fait de nouveaux intermédiaires nés avec l'Internet (Auto-by-tel, E-trade, Ebay, Priceline, Amazon). A contrario vouloir contourner les intermédiaires n'est pas toujours évident : par exemple, Levis vient d'abandonner la vente directe par le Net, sur son propre site.

technologies de l'information et de la communication. La diversité des situations de départ va engager les organisations sur des trajectoires cognitives et organisationnelles hétérogènes qui permettent d'interpréter des observations, au premier abord contradictoires (Eric Brousseau, 1997, p. 59). En voulant dépasser le débat récurrent entre déterminisme technologique et déterminisme organisationnel, Eric Brousseau et Alain Rallet (1997) adoptent une approche reposant sur une relation dynamique entre technologie et organisation formulée en termes de co-évolution : il existe une grande variété de sentiers de co-évolution qui restent en grande partie indéterminés, et qui doivent notamment tenir compte des effets de verrouillage, de la résistance au changement organisationnel et de l'intervention d'événements contingents. Différentes études montrent que les technologies de l'information et de la communication peuvent provoquer un rééquilibrage des pouvoirs au sein de l'organisation en facilitant à tous la mise à disposition de toutes les informations, mais également renforcer la conception hiérarchique et bureaucratique des organisations (Jean-Louis Peaucelle, 1997, p. 26-27). Eric Brousseau et Alain Rallet soulignent que les technologies de l'information et de la communication peuvent "constituer un facteur d'adoption d'architectures centralisées et spécialisées dans la mesure où elles en renforcent l'efficacité relative" (1997, p. 298)¹⁰. C'est particulièrement net dans le cas du déploiement des technologies Intranet, qui s'effectue rapidement dans les organisations. Si, en contribuant à déplacer l'emploi depuis les tâches répétitives, "automatisables", vers des tâches faisant plus largement appel à la capacité d'adaptation, de décision et de création de chacun, les technologies de l'information et de la communication peuvent augmenter l'autonomie et l'initiative des salariés, elles peuvent aussi permettre d'augmenter le contrôle et la surveillance (Daniel Kaplan, 1999). C'est le constat qu'effectuent M. Cézard, F. Dussert et M. Gollac (1992) qui au terme d'une analyse statistique de l'impact de l'informatique sur l'organisation (dans une étude intitulée "Taylor va au marché" !) concluent à la possibilité d'un "taylorisme flexible" aux antipodes des conceptions "postfordistes". Cette diversité peut même s'observer au niveau d'établissements identiques en terme d'activité, de taille, de technologies utilisées et d'environnement : l'étude de deux hypermarchés de taille moyenne et identique, situés dans des banlieues populaires de grandes villes françaises et appartenant à la même chaîne, montre comment la même exigence de

¹⁰ Eric Brousseau et Alain Rallet notent que cette tendance à contre courant du "discours dominant fondé sur les vertus des modèles décentralisés et intégrés" est confirmée par la "recentralisation des systèmes d'information à l'étape des réseaux" et "le caractère contre-productif de l'intégration des tâches à laquelle pousse une certaine utilisation de l'informatique (le cadre qui réalise des tâches de secrétariat, le chercheur qui se transforme en documentaliste...)" (1997, p. 298)

flexibilité peut être obtenue dans un établissement par une "gestion des ressources humaines taylorienne à très court terme" et dans l'autre établissement par "une gestion de la flexibilité par l'implication et la fidélisation du personnel" avec des modalités opposées point par point (Jean Gadrey, 1999 B, p. 154-155).

Une pluralité de cohérences possibles

Cette insistance sur la diversité des évolutions ne signifie pas que toutes les combinaisons de technologies, d'organisations et de cadres institutionnels soient possibles. Il est impératif que les différents aspects soient cohérents. Mais il existe une pluralité de cohérences possibles, chacune reposant sur une configuration particulière de conventions entre acteurs. Ces configurations peuvent être considérées comme des paradigmes socio-économiques en adoptant la signification de paradigme – formulée avec un autre objectif – de Michel Callon et Bruno Latour : "une certaine manière de concevoir et de percevoir le monde, arbitraire, cohérente et irréductible à toute autre (...), mais également une organisation sociale avec ses règles, ses formes de solidarité, d'apprentissage, la définition d'une identité propre" (1991, p. 18). Toutefois, il nous semble qu'au-delà de "l'assemblage d'une multiplicité d'éléments techniques, culturels, sociaux dont la seule association explique la robustesse" (idem, p. 20), la notion de paradigme avait été introduit par T.H. Kuhn dans l'histoire des sciences pour rendre compte également de l'existence temporaire d'un paradigme dominant, auquel peut se substituer un nouveau paradigme lors d'une révolution scientifique. Dans la mesure où le champ socio-économique se caractérise à l'inverse par la coexistence durable de plusieurs "paradigmes" différents, il nous semble plus pertinent d'utiliser le concept de "monde de production" (cf. chapitre VII).

Section II - L'aspect essentiel du logiciel au sein des technologies de l'information et de la communication

"La caractéristique clé des nouvelles technologies de l'information est leur programmabilité (...). Au risque de faire une tautologie, la programmabilité accroît l'importance des programmes, et des activités d'écriture des logiciels" (OCDE, 1993 B, p. 22). Pourtant l'importance du logiciel au sein des technologies de l'information et de la communication a été longtemps sous-estimée (A). Malgré les difficultés statistiques, il est

patent que les logiciels prennent de plus en plus d'importance, ce que nous appelons la tendance à la "logicialisation" de l'activité économique et sociale (B). Elle intègre le fait que l'importance des logiciels est également qualitative (et non seulement quantitative), ce qui confèrent aux logiciels un caractère stratégique (C).

A - UNE IMPORTANCE LONGTEMPS SOUS-ESTIMEE

La sous estimation de l'importance du logiciel renvoie aux visions "industrialistes" dominantes¹¹. Cette "industrialisation théorique des économistes" (Jean Gadrey) a des racines historiques profondes, et peut s'expliquer par les difficultés de mesure et d'intégration dans les principaux cadres théoriques existants, des activités non industrielles. C'est le cas des activités de services, où les aspects innovatifs, les gains de productivité (qui posent des problèmes particuliers de mesure) sont rarement pris en compte, quand ce n'est pas le caractère productif de ces activités qui est mis en cause.

Dans l'informatique la focalisation sur les aspects matériels au détriment des aspects immatériels¹², a pris des formes particulières. "L'ordinateur était le mythe révééré, (...), le logiciel et le service [faisant] figure de parent pauvre, simple moyen de promotion pour la vente de l'objet précieux : la machine" (Jean-Marie Desaintquentin, Bernard Sauteur, 1991, p. 49). Ce fétichisme de l'ordinateur était déjà présent dans le choix du terme d'ordinateur pour traduire "*electronic data processing machine*" : le traducteur avait retenu le mot "ordinateur" "parce qu'au Moyen Age, Dieu était le grand ordinateur, celui qui mettait de l'ordre dans le monde" (cité par Jean-Yvon Birrien, 1990, p.4). L'ordinateur était une "machine à fantasmés"¹³, ce dont témoignent les articles que la presse lui consacrait¹⁴. L'aspect "visible" de l'acquisition d'un ordinateur était un des facteurs explicatifs de la prédilection des

¹¹ Pour un florilège des conceptions "industrialistes" cf. Jean Gadrey (1999 D).

¹² Jusqu'à une période récente, la plupart des études économiques consacrées à l'informatique ne traitaient que de l'industrie du matériel informatique.

¹³ Tite d'un chapitre du livre de Jean-Pierre Brulé (1993).

¹⁴ Jean-Marie Desaintquentin et Bernard Sauteur (1991) dressent un florilège d'articles de presse particulièrement instructifs, intitulé "la fascination du mythe". L'anecdote suivante est révélatrice de la survalorisation de l'ordinateur en tant qu'objet matériel : lors de la première estimation électorale informatisée réalisée par RTL pour le référendum d'avril 1969, le journaliste commentait : "je suis dans un immeuble de verre et d'acier, avec des machines partout, une multitude de boutons qui clignotent et, devant ces machines, des hommes en blouse blanche", alors que l'ordinateur en question, un Honeywell 1200, comportait en tout et pour tout dix voyants qui ne clignotaient pas, et était utilisé par des informaticiens habillés "normalement" (p. 7).

décideurs pour ce type d'investissement et les aides massives (avec des succès mitigés) que l'industrie du matériel informatique recevait des pouvoirs publics. Par rapport à la fascination exercée par l'ordinateur, la conception des logiciels apparaissait comme une activité mineure, dont les difficultés étaient largement sous estimées¹⁵. Il est significatif qu'il n'ait jamais existé d'entreprises publiques dont l'activité principale était de produire des logiciels, y compris dans des pays comme la France où la place importante du secteur public s'est traduit quasi uniquement par des efforts au niveau du matériel (notamment le Plan Calcul) avec les tentatives récurrentes et volontaristes de constitution de grands producteurs d'ordinateurs nationaux à partir des entreprises publiques¹⁶. De ce fait, les seuls logiciels produits par des entreprises publiques le sont par des entreprises dont l'activité principale est autre (matériel informatique, télécommunications, aéronautique, espace, banques...). La seule tentative d'intervention directe de l'Etat français dans le domaine des logiciels fut la création de l'Agence de l'Informatique (ADI) en 1979 destinée à financer et à aider le développement de logiciels français dans les domaines de la bureautique (projet Kayak), des logiciels sectoriels destinés à une profession particulière (experts-comptables, avocats...), dans la création d'une alternative française à Unix (projet SOL), et dans le lancement de "Projets nationaux" concernant le génie logiciel, l'enseignement assisté par ordinateur, la traduction assistée par ordinateur... Les échecs de ces différentes tentatives conduiront à la dissolution de l'ADI en 1986¹⁷. Il est également significatif qu'il a fallu attendre 1993 pour que le "Manuel de

¹⁵ Le "Centre Mondial Informatique et Ressource humaine", créé par l'Etat en 1982 et présidé par Jean-Jacques Servan-Schreiber, constitue une caricature en la matière. Il s'était uniquement centré sur la question du matériel informatique en considérant que les logiciels seraient créés par les utilisateurs, "la programmation s'apprenant en huit jours". La vision de Thierry Saint Antoine, directeur du Centre mondial, était significative : "les passionnés qui fréquenteront les ateliers pourront créer des logiciels (...). A raison de trois logiciels créés par an et par atelier, on arrive à 150 000. Après, nous pourrions créer un centre mondial du logiciel" (cité par Jean-Pierre Brulé, 1993, p. 27-28). De façon moins anecdotique, il fallut beaucoup d'efforts pour faire reconnaître une science informatique à part entière, qui ne se réduisait pas à la fabrication de matériels.

¹⁶ Un bilan très critique de ces tentatives a été effectué Jean-Pierre Brulé, dans un ouvrage intitulé significativement, "L'informatique malade de l'Etat" et sous titré "Du plan calcul à Bull nationalisée : un fiasco de 40 milliards " (1993). Charles Goldfinger compare l'effort financier de quarante milliards de Francs consenti par les pouvoirs publics entre 1968 et 1993 en faveur de Bull, dont la "position concurrentielle, tant en France qu'à l'étranger, reste précaire", à l'absence de toute subvention de l'Etat pour Cap Gemini Sogeti, leader européen des services informatiques, réalisant plus de 80 % de son chiffre d'affaires à l'extérieur des frontières (1994, p. 63).

¹⁷ Ce constat ne doit toutefois pas masquer l'importance du rôle indirect de l'Etat dans l'économie des logiciels. Richard N Langlois et David C. Mowery (1996) soulignent le caractère déterminant des actions du gouvernement fédéral dans le développement de l'industrie américaine du logiciel : rôle des marchés publics notamment militaires (le Département de la Défense aux Etats-Unis est de très loin le principal consommateur de logiciels), des politiques de formation, d'aide à la recherche... En France, l'Etat représente près de la moitié (48 %) du chiffre d'affaires des SSII (données Syntec Informatique, citées in Abdelaziz Mouline, 1996, p. 24).

Frascati", dont l'objectif est de proposer un cadre méthodologique pour mesurer la recherche-développement, considère que certaines activités de création de logiciels peuvent être considérées comme de la recherche-développement (OCDE 1993 A).

Cependant un certain nombre d'évolutions récentes ont contribué à mieux prendre en compte l'importance du logiciel : le constat que si l'acquisition des équipements informatiques est de moins en moins coûteux, la mise en œuvre efficace de ces technologies paraît plus problématique (Pascal Petit, 1998, p. 18), les échecs de certains constructeurs informatiques et de certains grands projets (par exemple, les ordinateurs dits de "cinquième génération") en raison des difficultés dans le domaine des logiciels, le déplacement de la création de valeur et des marges dans la filière informatique, centré sur le *hardware* dans les années soixante-dix et quatre-vingt (modèle IBM et DEC), vers l'amont (les composants) et vers l'aval (logiciels et services) que symbolise le modèle Wintel, contraction de Windows-Microsoft et Intel (Eurostaf, 1995 B, p. 38-39)¹⁸.

B - LE CONSTAT STATISTIQUE : LA TENDANCE A LA "LOGICIALISATION" DE L'ACTIVITE ECONOMIQUE ET SOCIALE

1 - Malgré les difficultés d'évaluation, une évolution statistique incontestable

Une récente étude de l'OCDE (1998 A) consacrée au profil statistique du secteur du logiciel, souligne les difficultés pour appréhender cette activité. D'une part, il existe des problèmes de mesure pour l'ensemble de l'économie de l'information, qualifiée de "cible mouvante" : l'offre et la demande sont extrêmement mobiles, sous l'impulsion d'un changement technologique rapide et d'un marché international très dynamique ; les données quantitatives proviennent presque exclusivement de sources privées et sont "fréquemment défectueuses, quant à leur robustesse, au champ qu'elles couvrent ou à leur rigueur", et les statistiques officielles, "obtenues souvent avec retard, ne donnent qu'un aperçu assez sommaire" (OCDE, 1998 A, p. 5). D'autre part, le secteur du logiciel pose des problèmes spécifiques, (cf. l'"insaisissabilité" du logiciel analysée au chapitre III), qui se traduisent

¹⁸ Sont significatifs de cette évolution, les changements concernant les procédures d'agrément : dans les années soixante-dix et quatre-vingt, c'étaient les sociétés de services qui étaient des prestataires agréés par les constructeurs informatiques ; actuellement ce sont les constructeurs de matériels qui sont agréés par les éditeurs de progiciels. Il est également révélateur qu'un ouvrage récent intitulé "Techniques innovantes en informatique" (Luc Rubiello, 1997) soit exclusivement consacré au développement de logiciels.

notamment par les difficultés à saisir les frontières du secteur, en raison de la très grande variété des producteurs (David C. Mowery, 1996, p. 304).

De façon synthétique, l'OCDE estime que le logiciel représente environ 20 % du marché total des technologies de l'information, et qu'avec les services informatiques, souvent difficilement dissociables, son importance dépasse celle du matériel : la part du logiciel et des services informatiques au sein du marché mondial des technologies de l'information est passé de 32 % en 1980 à 46 % en 1987 et à 52 % en 1995¹⁹. Le logiciel est "un produit intermédiaire essentiel, et en forte croissance, pratiquement pour toutes les branches d'activités" (OCDE, 1998 A, p. 5). C'est le secteur des technologies de l'information et de la communication dont la croissance est la plus rapide (idem, p. 4).

Selon la Business Software Alliance (1999), les ventes de l'industrie américaine du logiciel ne cessent de s'accroître. Elles ont progressé de 17,8% en 1998, faisant passer la croissance annuelle moyenne du secteur de 10,6% durant la période 1990-1994 à 15,4% dans les années 1994-1998. Au total, le chiffre d'affaires du logiciel américain a atteint 140,9 milliards de dollars l'année dernière, tout juste devancé par l'automobile, et le secteur du logiciel devrait devenir le premier contributeur de l'économie américaine dès l'année 2000.

La croissance la plus rapide est celle des progiciels (cf. tableau et graphique). De 1985 à 1995, le taux de croissance annuel moyen des progiciels est de 16,3 % à comparer à un taux de croissance annuel moyen de 12,8 % pour l'ensemble des technologies de l'information. De ce fait, la part des progiciels passe de 13,5 % du marché mondial des TI en 1985 à 18,4 % en 1995.

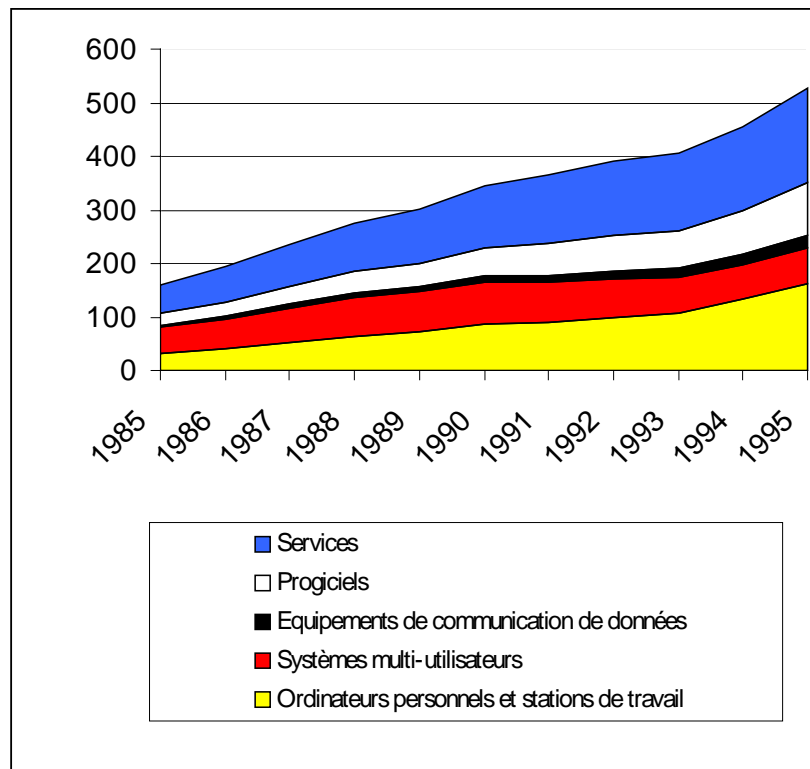
¹⁹ Il faut noter que, selon les sources, les chiffres peuvent être sensiblement différents : en 1996, Eurostaf estimait le marché mondial de l'informatique à 650 milliards de dollars qui se répartissaient en 440 milliards de dollars pour les logiciels et les services, et 210 milliards de dollars pour les matériels (Eurostaf, 1997 A, p. 5). Selon cette estimation, les logiciels et services représentaient donc 67 % du marché mondial de l'informatique en 1996 !

Tableau XV
Marché mondial des technologies de l'information par segment, 1985-95
(Pourcentages, sauf total mondial)

	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
Ordinateurs personnels et stations de travail	20,9	21,2	21,8	22,9	24,2	25	24,4	24,8	26,7	29	30,5
Systèmes multi-utilisateurs	29,5	28,7	28,1	26,8	25,3	22,9	20,7	18,8	16,3	14,3	13
Equipements de communication de données	3	2,9	2,9	2,9	3	3,1	3,3	3,6	4,1	4,3	4,3
Progiciels	13,5	13,8	13,9	14,2	14,4	15,5	16,6	17,4	17,6	17,9	18,4
Services	33,1	33,3	33,3	33,2	33,2	33,5	35,1	35,4	35,3	34,6	33,7
	100	100	100	100	100	100	100	100	100	100	100
Monde (en milliards de dollars)	158,3	193,5	235,1	276,6	300,3	345,4	364	391,6	405,1	455	527,9

Source : calculs d'après des données IDC, citées in OCDE, 1997 A, p. 16.

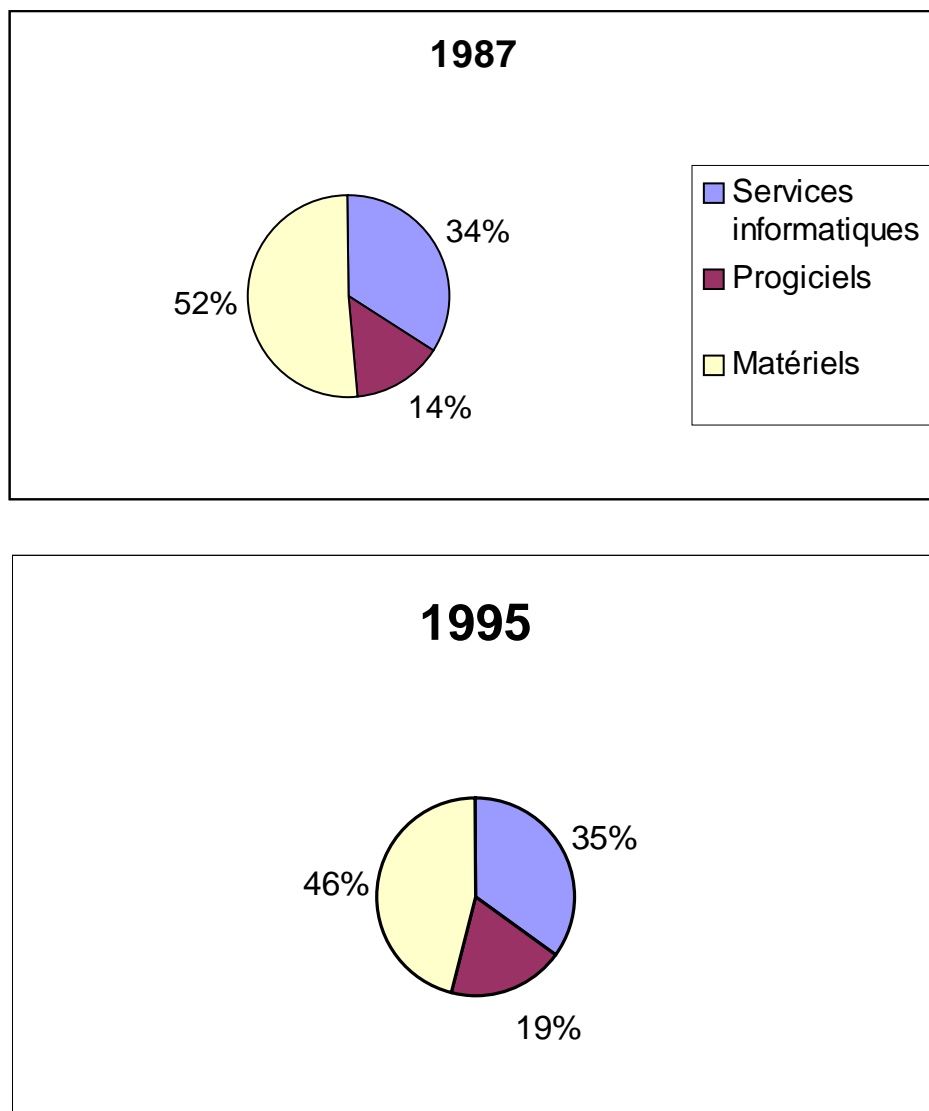
Graphique V
Marché mondial des technologies de l'information par segment, 1985-95
(en milliards de dollars courants)



Source : calculs et représentation effectués d'après des données IDC citées in OCDE, 1997 B, p. 251.

Pour les pays de l'OCDE uniquement (pays où les technologies de l'information sont les plus développées), l'importance prise par les progiciels est légèrement supérieure : la part des progiciels passe de 14,1 % en 1987 à 19,1 % en 1995 (cf. graphique). Sur cette période, le taux de croissance annuel moyen des progiciels (13,8 %) dépassait de 40 % celui de l'ensemble des technologies de l'information (9,7 %).

Graphiques VI
Répartition des technologies de l'information par segment,
(24 pays membres de l'OCDE)



Source : OCDE, 1997 B, p. 5.

Il faut noter que la part des progiciels est vraisemblablement sous-estimée pour plusieurs raisons. Premièrement, de par sa nature intangible, le progiciel est facile à dupliquer et à pirater, ce qui n'est évidemment pas le cas du matériel et des services (OCDE, 1998 A, p.

4). Deuxièmement, il existe dans le cas des progiciels une part non négligeable et croissante de production non marchande (les logiciels libres) qui sont fournis gratuitement ou vendus par des distributeurs à des prix qui ne couvrent pas les coûts de développement de ces logiciels. Troisièmement, les progiciels fournis par les constructeurs informatiques sans facturation séparée, ne sont pas comptabilisés. De plus, il existe des progiciels "indissociés", dont "le prix ne peut jamais être distingué de celui du matériel informatique avec lequel ils sont fournis" (Christine Afriat, 1992, p. 91). Ce phénomène, qui a toujours existé (par exemple dans les consoles de jeux), prend une importance grandissante au fur et à mesure que les frontières entre progiciels et matériel s'estompent : dans les réseaux de communication, les commutateurs logiciels remplacent les commutateurs matériels²⁰, et un éventail croissant de produits incorporent des progiciels (OCDE, 1998 A, p. 5). Il faut également tenir compte des progiciels qui sont fournis de façon intégrée avec des services informatiques (et éventuellement du matériel) dans le cadre de ce que l'on peut appeler un "marché des solutions informatiques" (Christine Babelon, 1987, p. 173).

Si la composante la plus dynamique du secteur des logiciels est sans conteste les progiciels, il faut également intégrer les logiciels sur mesure. Ceux-ci présentent deux difficultés sur le plan statistique : ils sont le plus souvent effectués dans le cadre plus général d'une prestation de service informatique ; une part importante est effectuée en interne. Les logiciels constituent le plus souvent une sous-catégorie des services informatiques, mais il importe de noter que la plupart des autres prestations de services informatiques intègrent, de façon variable et difficilement quantifiable, une part de réalisation de logiciels²¹.

La notion de services informatiques désigne l'ensemble des prestations liées à l'étude, à la mise en place, à l'utilisation, à l'exploitation et à la maintenance des systèmes et matériels informatiques. Chacune des sources d'information établit ses statistiques sur la base de définitions qui lui sont propres, et à partir de champs d'études qui se recouvrent plus ou moins. Il existe ainsi plusieurs classifications internationales officielles et des classifications nationales différentes selon les pays²². Il faut de plus prendre en compte les nombreuses

²⁰ Pour les télécommunications et les systèmes de production automatisée, le logiciel peut atteindre la moitié du coût total (Michel Delapierre, Jean-Benoît Zimmermann, 1984, p. 11).

²¹ Comme le souligne Gérard Dréan (1996 A, p. 195), "de nombreux services consistent en grande partie en développement de logiciels".

²² Une comparaison de ces différentes classifications est effectuée dans OCDE (1998 A, p. 7-12).

sources privées qui ont leurs propres classifications : IDC, Gartner Group, INPUT, Datamation et, pour la France, PAC (Pierre Audoin Conseil) et Syntec Informatique. De plus les différentes classifications sont fréquemment modifiées pour tenir compte de la rapidité des évolutions caractérisée par l'apparition et la disparition périodiques de certaines catégories de services. Par exemple l'INSEE qui recense les services informatiques depuis 1981, les avait classés dans deux secteurs de la NAP 600 : le 7703 (études informatiques et d'organisation) qui comprenait les sociétés d'études informatiques mais également les sociétés de conseil en organisation, et le 7704 (travaux à façon informatique), dont l'importance a considérablement diminué. Dans la nouvelle NAF établie en 1993, les services informatiques sont regroupés seuls dans une seule catégorie (72 Activités informatiques) avec un découpage interne sensiblement différent (cf. tableau).

Tableau XVI
Comparaison des deux nomenclatures de l'INSEE pour les services informatiques.

NAP 3000		NAF	
7703	Etudes informatiques et d'organisation	72	Activités informatiques
77031	<i>Conseil de direction, d'organisation, de gestion</i>	721	Conseil en systèmes informatiques
77032	<i>Formation du personnel des entreprises</i>	722	Réalisation de logiciels
77033	<i>Conseil en recrutement, sélection du personnel</i>	723	Traitement de données
77034	<i>Organisation, formation, recrutement</i>	724	Activités de banques de données
77035	Etudes en informatique, logiciels non standard	725	Entretien et réparation de machines de bureau et de matériel informatique
77036	Ventes ou location de logiciels	726	Autres activités rattachées à l'informatique
77037	Services informatiques		
7704	Travaux à façon informatique		
77041	Travaux à façon informatique, saisie de données		
77042	Locations d'heures machine		

La classification sectorielle pose également des problèmes en raison des fréquentes réorientations de l'activité économique dans ce secteur : changement des activités de base menées par les établissements d'une entreprise, modification de la structure organisationnelle de l'entreprise (fusions et acquisitions, démembrements) (OCDE, 1998 A, p. 15). Enfin il faut noter que beaucoup de données privées sont révisées a posteriori ce qui rend les séries chronologiques peu fiables : par exemple en 1993, IDC estimait le marché mondial des progiciels de 1991 à 51,05 milliards de dollars (à prix courants) ; en 1995 une nouvelle estimation d'IDC évaluait le marché mondial des progiciels de 1991 à plus de 60 milliards de dollars à prix courants (OCDE, 1998 A, p. 27).

En 1992 le marché mondial des services informatiques se répartissait ainsi :

Tableau XVII
Marché mondial des services informatiques

	en milliards de dollars	en pourcentage
Traitements	36	15,9%
Réseaux	20	8,8%
Intégration de systèmes	16	7,0%
Organisation de systèmes	11	4,8%
Prestations intellectuelles	49	21,6%
Formation	7	3,1%
Conseil	10	4,4%
Logiciels outils de base	40	17,6%
Logiciels d'application	38	16,7%
Total	227	100,0%

Source : calculs effectués d'après les données de Datamation, citées in Eurostaf (1995 B, p. 21)

Entre 1990 et 1995, le marché américain des logiciels et services informatiques a connu un taux de croissance annuel moyen de 12 %, et se répartissait ainsi :

Tableau XVIII
Marché des services informatiques des Etats-Unis
(en millions de dollars courants)

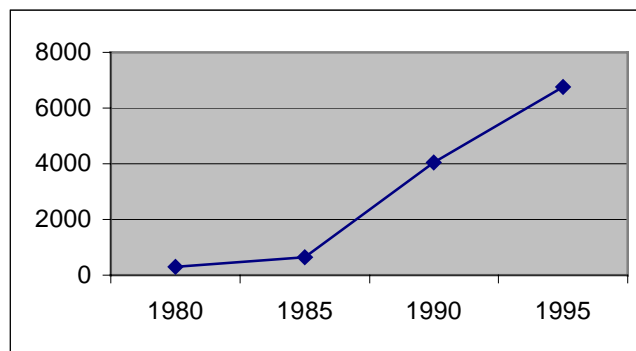
	1990	1995	Taux de croissance global	Taux de croissance annuel moyen
737 Services de programmation informatique, traitement de données, et autres activités de services informatiques rattachées	88 299	152 213	72%	12%
7371 Services de programmation informatique	21318	37447	76%	12%
7372 Progiciels	16523	31087	88%	13%
7373 Configuration de systèmes informatiques intégrés	12916	20592	59%	10%
7374 Préparation et traitement de données, services de traitement	17820	31144	75%	12%
7375 Services de récupération d'information	3547	5489	55%	9%
7376 Services de sous traitance informatique	1994	3110	56%	9%
7377 Location et leasing informatiques	2644	2213	-16%	-3%
7378 Maintenance et réparation informatique	7000	8228	18%	3%
7379 Services informatiques n.c.a.	4537	12903	184%	23%

Source : calculs effectués d'après les données de l'OCDE (1998 A, p. 23)

Sur une période de 10 ans, le taux de croissance annuel moyen de l'ensemble du secteur logiciel et services informatiques aux Etats-Unis a dépassé 13 %. Ce secteur qui représentait 1,1 % du PIB américain en 1958, en représente 2,7 % en 1995 (d'après les données OCDE, 1998 A, p. 18).

Au Japon, pour la même période, la croissance est encore plus rapide (le taux de croissance annuel moyen dépasse 26 % entre 1985 et 1995, et il atteint même 44 % sur la période 1985-1990), mais le poids du secteur dans l'économie japonaise demeure plus faible qu'aux Etats-Unis : il passe de 0,5 % du PIB en 1985 à 1,3 % en 1995 (d'après les données OCDE, 1998 A, p. 18).

Graphique VII
Logiciels et services informatiques au Japon
(en millions de dollars courants)



Source : représentation d'après les données de l'OCDE (1998 A, p. 18)

Pour le marché européen, l'évolution est la suivante :

Tableau XIX
Les différents segments du marché européen
(en milliards de dollars courants)

	1993	1994	taux de croissance 94/93	1999 (prévision)	TCAM 99/93 (prévision)
Prestations intellectuelles	20,5	21,0	2,4%	25,0	3,4%
Prestations matérielles	19,4	19,2	-1,0%	19,1	-0,3%
Logiciels systèmes	11,8	12,2	3,4%	14,5	3,5%
Systèmes clé en main	11,1	11,7	5,4%	16,6	6,9%
Logiciels applicatifs	9,3	10,1	8,6%	17,2	10,8%
Services de traitement	7,3	7,6	4,1%	9,0	3,6%
Services réseaux	4,9	5,6	14,3%	11,6	15,4%
Intégration de systèmes	4,1	4,6	12,2%	8,0	11,8%
Infogérance	2,5	3,2	28,0%	7,65	20,5%
Total	90,9	95,2	4,7%	128,65	6,0%

*Source : calculs effectués d'après les données Input,
citées in Eurostaf (1996 A, p. 25-26)*

Globalement, à partir des données de l'EITO (European Information Technology Observatory), la croissance du secteur du logiciel et des services en Europe serait de 22,1 % entre 1992 et 1996, soit près du double de celle du matériel (11,7 %). On constate que de même que le niveau d'équipement en matériel informatique reflète l'état de développement d'un pays, la part des logiciels et des services dans les technologies de l'information augmente avec le niveau de développement des pays. Par exemple, le marché des technologies de l'information en Chine en 1994 se répartissait ainsi :

Tableau XX
Répartition du marché des technologies de l'information en Chine
(en pourcentage)

Matériel	Logiciel	Communication de données	Services
87 %	3 %	8 %	2 %

Source : IDC et OCDE, citées in Eurostaf (1996 A, p. 24)

Il faut toutefois noter qu'à la même époque, le taux de piratage des progiciels en Chine était estimé à plus de 90 % !

En France selon Pierre Audoin Conseil, le secteur logiciels et services avait une croissance annuelle moyenne de 4,9 % entre 1991 et 1995 alors que les dépenses en matériel en 1995 étaient au même niveau qu'en 1991 (Eurostaf 1996 A, p. 206).

A partir de la nomenclature de l'INSEE, le chiffre d'affaires du secteur logiciels et services se répartissait ainsi :

Tableau XXI
Répartition du secteur logiciels et services en France
(chiffre d'affaires 1995 en millions de dollars courants)

		Chiffre d'affaires	Répartition 1995	Répartition 1996
72	Activités informatiques	25195	100%	100%
721	Conseil en systèmes informatiques	7387	29%	30%
722	Réalisation de logiciels	7488	30%	32%
723	Traitement de données	7515	30%	30%
724	Activités de banques de données	487	2%	2%
725	Entretien et réparation de machines de bureau et de matériel informatique	2318	9%	6%

*Source : données OCDE (1998, p. 23)
et INSEE (Secrétariat d'Etat à l'Industrie, 1999, p. 28).*

En termes d'emploi, l'évolution était la suivante :

Tableau XXII
Evolution des effectifs salariés en France

72	Activités informatiques	1981	1985	1989	1993	1996	TCAM 96/81
721	Conseil en systèmes informatiques	62532	100192	135216	148216	169156	6,9%
722	Réalisation de logiciels	5283	10938	21285	33639	52116	16,5%
723	Traitement de données	6197	14297	30540	45837	55837	15,8%
724	Activités de banques de données	27285	35110	41333	43232	40904	2,7%
725	Entretien et réparation de machines de bureau et de matériel informatique	1472	2477	3935	4255	4985	8,5%
726	Autres activités rattachées à l'informatique	22272	37359	38065	20918	14527	-2,8%

Source : calculs effectués d'après des données UNEDIC in DARES (1998, p. 4).

On dispose également pour la France des données Input, déjà utilisées au niveau européen :

Tableau XXIII
Les différents segments du marché français
(en milliards de Francs courants)

	1994	1995	Taux de croissance 95/94	2000 (prévision)	TCAM 2000/94 (prévision)
Prestations intellectuelles	38,8	39,4	1,5%	42,4	1,5%
Prestations matérielles	19,9	19,5	-2,0%	19,7	-0,2%
Logiciels systèmes	15,8	16,5	4,4%	19,3	3,4%
Systèmes clé en main	11	11,4	3,6%	15,8	6,2%
Logiciels applicatifs	14,9	15,6	4,7%	23,1	7,6%
Services de traitement	8,9	8,9	0,0%	10	2,0%
Services réseaux	8,3	9,1	9,6%	18,8	14,6%
Intégration de systèmes	5,4	5,8	7,4%	9,6	10,1%
Infogérance	4,7	5,5	17,0%	12,2	17,2%
Total	127,7	131,7	3,1%	170,9	5,0%

Source : calculs effectués d'après les données Input, citées in Abdelaziz Mouline (1996, p. 23)

On constate que la structure et les évolutions du marché français sont peu différentes du constat effectué au niveau européen.

Enfin, pour la France, on dispose également des données fournies par Pierre Audoin Conseil.

Tableau XXIV
Croissance annuelle des différents segments du marché français
(en %, sauf C.A. 1996 en millions de F.)

	1991	1992	1993	1995	1996	Chiffre d'affaires 1996
Progiciels	13,8	10,0	7,5	9,9	9,0	25,73
Conseil, Assistance, Réalisation	6,5	5,5		7,1	6,0	24,66
Maintenance matériel	2,4	-1,0	-2,5	-5,6	-2,9	14,41
Logiciel de base	10,0	6,5	2,0	2,0	3,0	12,63
Intégration de systèmes	12,0	11,0	4,5	7,5	9,0	11,56
Infogérance				13,7	13,0	11,67
Traitement	10,0	8,7	12	8,8	10,0	9,17
Ingénierie de solutions	4,7	5,3		6,2	6,0	9,29
Total	8,0	5,8	3,0	5,8	6,3	119,13

Source : d'après les données de Pierre Audoin Conseil, citées in Eurostaf, 1995 B p. 23 et Eurostaf 1997 B p. 246

Il importe de remarquer que toutes ces statistiques ne concernent que la production primaire de logiciels et de services informatiques, c'est à dire la production vendue sur le marché. La production totale de logiciels et services informatiques intègre également la production secondaire, c'est à dire les logiciels et les services informatiques réalisés par entreprises n'appartenant pas au secteur informatique, comme les cabinets d'experts-comptables ou les sociétés de conseil, ainsi que la production réalisée par les organisations utilisatrices (par exemple, le système de réservations aériennes Sabre d'American Airlines). En général, les statistiques publiques ou privées n'en tiennent pas compte alors que cette production est importante. Par exemple, en France le ratio entre logiciels produits pour

compte propre et investissements totaux en logiciels était estimé à 42 % en moyenne pour les années 1985-1990²³ (OCDE, 1998 A, p. 6).

Toutefois, lorsque la formation brute de capital fixe a été étendue aux logiciels lors de la mise en application du nouveau système européen de comptabilité nationale (SEC 95), a été définie une FBCF en logiciels qui prend en compte, en plus de l'acquisition des logiciels spécifiques et standards, les logiciels développés pour compte propre. Par contre la FBCF en logiciels n'intègre pas les logiciels fournis avec le matériel, comptabilisés en FBCF de matériel, ni les achats par les ménages, consommation qui reste encore faible en regard des achats des entreprises et des administrations.

Tableau XXV
Evolution de la FBCF en logiciels en France
(en millions de F. courants)

	1992	1993	1994	1995	1996
FBCF en logiciels	32 226	30 082	31 318	36 092	43 241
Taux de croissance		-6,65%	4,11%	15,24%	19,81%

Source : Données sur la FBCF issues des Comptes nationaux, citées par Jean-Pierre Berthier (1999, p. 12)

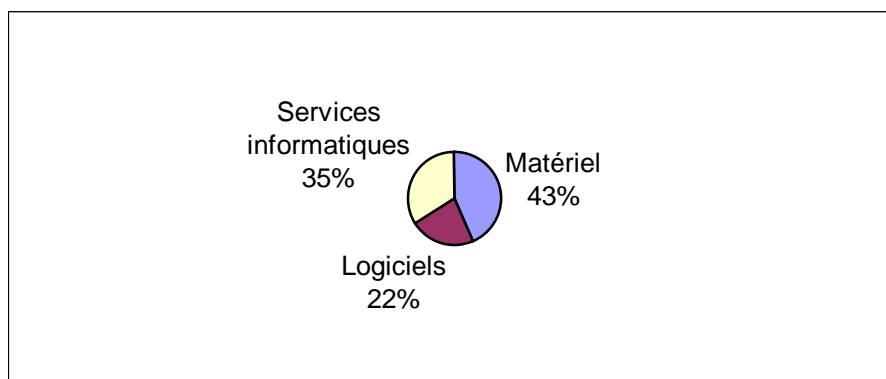
La forte croissance de la FBCF en logiciels (7,63 % de taux de croissance annuel moyen sur la période 1992-1996) confirme l'importance des logiciels dans les investissements immatériels mise en évidence dans des études plus anciennes. Une étude du Crédit National montrait que de 1974 à 1983, au sein de l'investissement immatériel lui-même en forte croissance, les logiciels étaient la seule composante dont la part relative avait augmenté (Christine Afriat, Pierre Caspar , 1988, p. 20) ; en 1983, l'investissement logiciel interne était estimé légèrement inférieur à l'investissement logiciel externe (idem, p. 63).

De même, certaines études des dépenses informatiques des entreprises intègrent pour les logiciels, les dépenses liées à la création de logiciels en interne. C'est le cas de l'étude du

²³ Une façon indirecte de mesure l'ampleur de cette production est le constat qu'en mars 1997, en France, sur 281 000 informaticiens, 120 000 travaillaient dans le secteur des technologies de l'information et de la communication, et 161 000 dans d'autres domaines d'activités (Secrétariat d'Etat à l'Industrie, 1999, p. 162).

Secrétariat d'Etat à l'Industrie, sur les dépenses informatiques des entreprises industrielles (à noter par contre que les dépenses de logiciels ne comprennent, pour les acquisitions de logiciels, que celles qui sont dissociées du matériel informatique) :

Graphique VIII
Répartition des dépenses informatiques des entreprises industrielles
(France, 1997)



Source : *Enquête annuelle d'entreprise, Secrétariat d'Etat à l'Industrie (1999, p. 72)*

Aux Etats-Unis, l'investissement en logiciels, intégrant les logiciels développés pour compte propre, a été estimé par le US Bureau of Economic Analysis (1997).

Tableau XXVI
Initial Estimates of Investment in Computer Software
(en milliards de dollars courants)

	1987	1992	TCAM 1987-1992
Computer programming services	13,5	25	13,1%
Prepackaged software	10,5	21,2	15,1%
Own-account software production	51,4	81	9,5%
Investment in computer software	78,4	125,8	9,9%

Source : *calculs effectués d'après les données US Bureau of Economic Analysis (1997, p. 3).*

On constate la forte croissance sur la période de l'investissement en logiciels (près de 10 % de taux de croissance annuel moyen) dont la composante la plus dynamique est constituée par les achats de progiciels. Cette forte croissance a fait passer les investissements en logiciels de 1,67 % du PIB américain en 1987 à 2,01 % en 1992. Il faut noter que si les logiciels développés pour compte propre ont vu leur importance diminuer, ils continuent à constituer la part la plus importante des investissements en logiciels.

De façon plus générale, à partir de la connaissance des dépenses de recherche-développement en matière de logiciels pour certains pays, l'OCDE en déduit que "l'utilisation totale de logiciels par les entreprises est très supérieure à ce qu'indique seulement le chiffre d'affaires du marché" (OCDE, 1997 B, p. 68).

Le dernier problème statistique, pour apprécier l'importance acquise par les logiciels au sein de l'informatique et plus généralement de l'activité économique, concerne la nécessaire prise en compte de l'évolution des prix. Toutes les données que nous avons citées sont en monnaie courante. La mesure de la variation des prix des logiciels est particulièrement délicate. Elle combine trois problèmes qui renvoient à la nature particulière du produit logiciel. Le premier problème, qui existe également pour le matériel informatique, a trait aux différences selon les méthodes utilisées pour traiter l'effet qualité concernant des produits dont les caractéristiques changent rapidement (traitement par chaînage, méthodes hédoniques...). Le second problème plus spécifique au logiciel sur mesure, que l'on retrouve pour de nombreuses activités de services, concerne l'évaluation de la personnalisation du produit. Le troisième problème, commun aux biens intangibles, concerne la faiblesse des coûts de reproduction, qui tendent vers zéro avec les possibilités de livraison électronique. S'y ajoute dans le cas des progiciels, les stratégies particulières liées à la force des rendements croissants d'adoption et notamment des externalités de réseaux directes et indirectes. Il existe peu d'études sur la question. Une étude de la Software Publishers Association basée sur la méthode des modèles correspondants et portant sur les logiciels d'application pour ordinateur personnel en Amérique du Nord, estime à -2,7 %, le taux de diminution annuel moyen des prix des logiciels sur la période 1987-1993. Cette baisse est variable selon le type de logiciel : elle est de -1,1 % pour les traitements de texte, de -4 % pour les tableurs et de -4,2 % pour les systèmes de gestion de bases de données. Pour ce dernier type de logiciels, une autre étude menée en Allemagne entre 1986 et 1994 estimait la baisse annuelle moyenne des prix à -7,4

% par la méthode hédonique et à -9,3 % par la méthode des modèles correspondants²⁴ (OCDE, 1998 A, p. 41-42).

Certes cette baisse des prix est moins rapide que dans le matériel informatique, mais elle est par contre vraisemblablement plus rapide que dans la partie "services purs" des services informatiques. Le déplacement de la valeur ajoutée du matériel vers le logiciel est illustré par la boutade suivante du dirigeant d'un grand constructeur américain : "de 1950 à 1960, le matériel était vendu, le logiciel était donné ; de 1960 à 1990 le matériel et le logiciel sont vendus séparément ; à partir de 1990 le logiciel sera vendu et le matériel donné" (cité par Economie et prévision, 1986). Si ce pronostic ne s'est pas complètement réalisé²⁵, certains auteurs estiment que l'on assiste à une inversion de la règle des 80-20 qui caractérisait les parts relatives du matériel et du logiciel à l'ère de la "grande informatique" (Jean-Benoît Zimmermann, 1995, p. 183, David Mowery, 1996, p. 3).

L'importance acquise par le logiciel est illustrée par le fait que c'est un éditeur de progiciels (Microsoft) qui est devenu la première capitalisation boursière mondiale. La mutation d'IBM est également significative de cette évolution. IBM était essentiellement à l'origine un producteur de matériels informatiques. Elle domina longtemps de façon hégémonique l'informatique mondiale, et elle représente toujours le chiffre d'affaires le plus important dans l'informatique. Depuis 1995, la part du logiciel et des services a dépassé le matériel dans les ventes d'IBM, et son objectif est de réaliser les deux tiers de son chiffre d'affaires dans les logiciels et les services pour la fin de la décennie (Eurostaf, 1996 A, p. 136).

Cette évolution nous conduit à parler d'une "logicialisation" de l'activité économique et sociale.

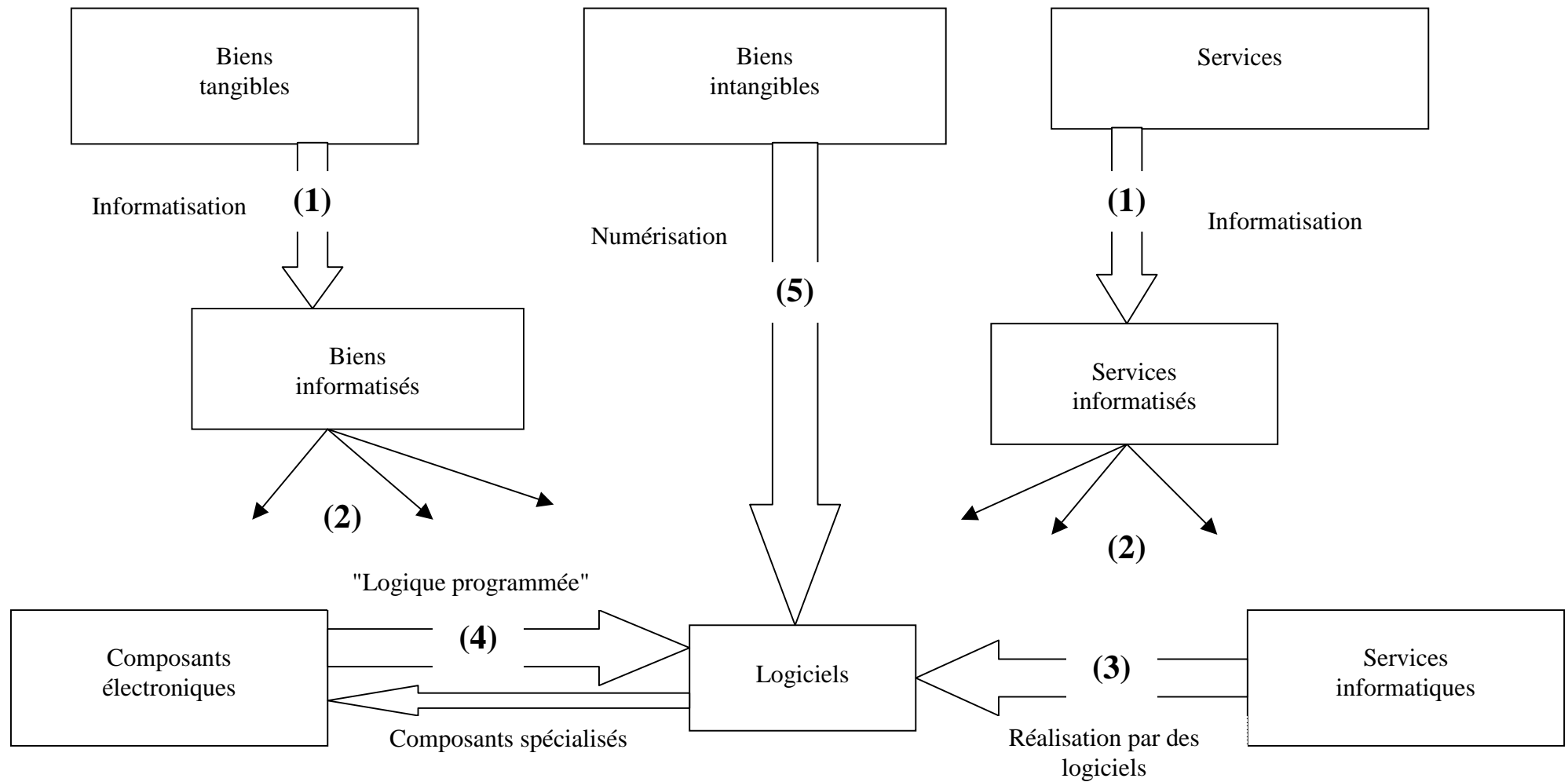
²⁴ La baisse plus rapide estimée par la méthode des modèles correspondants peut s'expliquer par la baisse forte des prix des anciennes versions d'un progiciel fréquemment observée en cas de commercialisation simultanée d'une nouvelle version.

²⁵ Si ce sont développées récemment quelques initiatives spectaculaires de fourniture gratuite d'ordinateurs personnels, elles sont liées à la consommation de certains services (par exemple d'accès à Internet) ou à un financement indirect par de la publicité (voire dans certains cas, à l'acquisition de données fines sur les utilisateurs, permettant de constituer des "profils" valorisables ultérieurement). Le modèle (matériel donné, services facturés) fit le succès du Minitel en France. Il faut toutefois prendre en compte, dans le cas du micro-ordinateur, le fait qu'il peut être utilisé pour d'autres usages que le service facturé lié, et il est prématuré pour se prononcer sur la viabilité économique à terme de ce type de pratiques.

2 - La tendance à la "logicialisation" de l'activité économique et sociale

Nous définissons la "logicialisation" de l'activité économique et sociale comme étant la combinaison de deux phénomènes : l'informatisation des biens et des services qui entraîne le développement du secteur informatique (matériels, logiciels et services informatiques) ; au sein du secteur informatique, la part de plus en plus importante prise par le logiciel au détriment du matériel et des services informatiques. La "logicialisation" intègre également un phénomène plus récent mais dont la croissance est très rapide, à savoir la numérisation progressive de l'ensemble des biens intangibles et ses conséquences (cf. schéma IX).

Schéma IX : La "logicalisation" de l'activité économique



En suivant les flux du schéma IX, on peut effectuer les remarques suivantes. L'informatisation des biens et des services (1) a été déjà largement analysée. L'informatisation des biens consiste à doter les biens de fonctions informatives, de possibilités de programmation par l'utilisateur et de réaction à l'environnement, de plus en plus sophistiquées. Elle concerne une part croissante des biens de consommation, et la majorité des biens d'équipements (Machines Outils à Commande Numérique, automates programmables, robots...). L'informatisation de services de plus en plus divers se développe rapidement, comme en témoigne le fait que c'est le secteur qui est le principal utilisateur des technologies de l'information dans toutes les économies développées (OCDE, 1993, p.14)²⁶ avec notamment le développement de nombreuses situations de self-service (Jonathan Gershuny, 1978)²⁷. La forte croissance des biens et des services informatisés, à laquelle il faut intégrer la création de nouveaux biens et services d'emblée informatisés, a induit une demande importante en matériel informatique divers, en composants électroniques, en logiciels et en services informatiques (2).

A l'intérieur de ce secteur informatique en forte croissance, on constate statistiquement que la part des logiciels augmente. La première explication est qu'une proportion croissante des services informatiques est réalisée par des logiciels (3). C'est notamment le cas pour les services de support (formation, aide à l'installation et à l'utilisation, maintenance et gestion de systèmes...), ce qui expliquerait que ces services n'aient pas connu la croissance exponentielle que prévoyaient ceux qui se fondaient uniquement sur la croissance des besoins de support dus à l'existence de nouveaux produits et de nouveaux usages, en lien avec l'impressionnante augmentation de la puissance informatique installée (Gérard Dréan, 1996 A, p. 279-280).

La deuxième explication est qu'une part croissante des traitements informatiques qui étaient effectués directement par des composants matériels sont réalisés par des logiciels (4). Il faut à ce niveau prendre en compte le fait que, techniquement, les possibilités de

²⁶ Ce même rapport souligne qu' " il est donc paradoxal que la plupart des travaux statistiques détaillés sur les technologies de l'information soient toujours axés principalement sur l'industrie", alors que les logiciels et services informatiques représentent, à eux seuls, plus du quart des dépenses de recherche-développement des entreprises du secteur des services des pays de l'OCDE (OCDE, 1998, p. 4). De même, plusieurs études convergentes indiquent que "les services regroupent 85 % du total des investissements privés dans les technologies de l'information" (Jean Gadrey, 1996, p. 150).

²⁷ Il faut toutefois noter que de même que l'automatisation du travail industriel a révélé l'importance de certains aspects du travail qui n'étaient pas formellement reconnus, l'automatisation de certains services dévoile l'utilité de certaines dimensions des services, notamment relationnelles, quand elles sont supprimées.

substitution entre composants matériels et composants logiciels existent dans les deux sens, et il existe des exemples d'opérations logicielles qui en étant gravées dans le silicium sont transformées en composants matériels : par exemple, les premiers micro-ordinateurs comprenaient fréquemment un interpréteur BASIC gravé dans la mémoire morte (ROM) de l'ordinateur ; récemment IBM a développé un processeur spécialisé pour protéger la confidentialité des documents contenus sur un ordinateur ou envoyés sur Internet, aspect décisif notamment pour le développement du commerce électronique, alors que les fonctions d'encryptage des fichiers et de signature électronique pour authentifier l'auteur ou l'émetteur d'un document sont habituellement réalisées par des dispositifs logiciels (18h.com Le quotidien de l'Expansion, 27/9/99). Un exemple intéressant est le choix opéré par les concepteurs du Mac de placer l'interface graphique dans la mémoire morte de la machine, ce qui incitait fortement les développeurs d'application à utiliser cette interface plutôt que d'en construire une spécifique à l'application (Frederick P. Brooks, 1996, p. 229-230). Ceci a contribué à la facilité d'utilisation de cette interface devenue standard pour toutes les applications Mac et donc à la facilité d'utilisation du Mac qui contribua grandement à son succès. Mais ceci s'avéra également un sérieux handicap quand se développa une interface graphique concurrente (Windows) avec la nécessité pour les développeurs d'applications d'effectuer un double développement de leurs applications.

Les considérations économiques expliquent qu'au-delà de ces cas particuliers, c'est la substitution du logiciel au matériel qui l'a largement emporté sur l'opération inverse : plus grande flexibilité de ce qui est exécuté par des logiciels que par des dispositifs matériels²⁸, ce qui est un atout décisif vu la rapidité des évolutions et des corrections nécessaires, possibilité de fabriquer des composants matériels moins spécialisés source d'économies d'échelle déterminantes dans ce secteur. C'est ce que Jean-Benoît Zimmermann (1995, p. 183) appelle le passage d'une "logique câblée à une logique programmée". Une illustration de ce phénomène est l'introduction dans les années quatre-vingt et le rapide développement au début des années quatre-vingt-dix, des processeurs à architecture RISC (*Reduced Instruction Set Computing*). A la différence des processeurs CISC, les processeurs RISC ont un répertoire d'instructions exécutables par le matériel extrêmement réduit, ce qui permet des gains en

²⁸ C'est ce qui explique que la tentative de Digital Research de faire graver par Intel son système d'exploitation CP/M (l'un des premiers systèmes d'exploitation pour micro-ordinateurs) n'ait pas abouti et n'a jamais été reprise par un autre producteur de logiciel par la suite.

termes de coût et de rapidité, toutes les opérations plus complexes étant réalisées par une "couche" supplémentaire de logiciels. Il est vraisemblable que la même évolution s'opérera, pour les mêmes raisons (flexibilité et économies d'échelle), pour l'ensemble des biens qui contiennent à l'heure actuelle des composants électroniques spécifiques, dédiés à un usage particulier, et qui, au fur et à mesure de l'augmentation de leur puissance de traitement et de leur interconnexion, seront progressivement remplacés par des composants électroniques universels, la différenciation de leurs fonctions étant assurée par des logiciels (Jean-Benoît Zimmermann, 1998 A, p. 131-132).

La dernière dimension de la "logicialisation" concerne les conséquences de la numérisation, qui se généralise rapidement, de l'ensemble des biens intangibles quelle que soit leur forme originelle (écrit, image, photo, son, film...). D'une part, la production, la diffusion, la consommation de ces biens nécessite du matériel adéquat mais également de plus en plus de logiciels²⁹ (cf. par exemple l'importance des logiciels de compression de données³⁰). D'autre part, les possibilités accrues d'interactivité avec les utilisateurs et d'intégration de documents existants à l'origine sous des formes très variées, offrent de nouvelles opportunités de création, notamment culturelles, et estompent la distinction avec la création de logiciels (5).

Les logiciels vont donc certainement continuer à voir leur poids augmenter dans l'activité économique. L'OCDE considère qu'il est déjà le "secteur le plus important économiquement, et qu'il est à l'économie fondée sur le savoir ce que les secteurs de l'acier et de l'automobile étaient à l'économie industrielle" (1998 A, p. 4). Mais au-delà de ces aspects quantitatifs, c'est également sur un plan qualitatif que l'importance des logiciels est croissante.

C - L'IMPORTANCE QUALITATIVE DES LOGICIELS

C'est tout d'abord par rapport à l'ensemble de l'informatique que les logiciels jouent un rôle qualitativement de plus en plus important. Certes il existe des relations de complémentarité techniques mais également économiques entre le matériel et le logiciel,

²⁹ Asdrad Torrès souligne que les "principaux bénéficiaires du multimédia pourraient bien être (...) les entreprises de logiciels" (1996, p.16).

³⁰ Par exemple, le développement de la vidéo numérique ne peut s'effectuer qu'à l'aide des logiciels de compression, qui permettaient en 1995 de réduire les 45 millions de bits par seconde à 1,2 (Nicholas Negroponte, 1995, p 30).

comme l'illustre la course-poursuite entre matériels de plus en plus puissants et logiciels de plus en plus "gourmands". Aux origines de l'informatique, c'était le matériel qui était la question la plus importante, ce qu'atteste le poids qu'avaient les constructeurs informatiques sur l'évolution de l'ensemble du secteur. C'est avec l'irruption de la micro-informatique qu'un basculement décisif se produisit. Les succès initiaux de la micro-informatique sont largement dus à un logiciel (Visicalc, premier tableur en 1979) et la forte croissance ultérieure de la micro-informatique doit beaucoup aux services de plus en plus diversifiés qu'offre un micro-ordinateur grâce à une gamme de plus en plus étendue de logiciels dans les domaines les plus divers. Il est significatif qu'à cette époque la question des standards décisifs en informatique se soit déplacée du matériel vers les logiciels (cf. chapitre II) et que les producteurs de logiciels soient devenus les acteurs déterminants de l'évolution de la filière.

En même temps que le matériel se banalise, les logiciels, qui confèrent au matériel son utilité³¹, occupent une place de plus en plus stratégique au sein de l'informatique. C'est la conclusion à laquelle parviennent différentes études sur le sujet, dont l'évolution est éloquente, les plus anciennes mentionnant également parfois les composants et les services comme éléments déterminants : "deux des nœuds technologiques les plus importants de la filière électronique sont les composants électroniques et les logiciels qui représentent une partie des clés de la domination de la filière" (Eric Labat, 1984, p.15) ; "le logiciel représente à la fois l'enjeu majeur de l'industrie informatique et son principal goulot d'étranglement" (Christine Babelon, 1987, p. 21) ; "le logiciel est l'élément essentiel des systèmes de technologies de l'information et la principale source de coûts" (OCDE, 1991, p. 9) ; "le logiciel et les services sont les moteurs de la croissance" (Daniel Weissberg, 1992, p.73) ; "plus que jamais l'évolution [du logiciel] est au centre de toute l'industrie informatique" (Gérard Dréan, 1996 A, p. 255) ; "le logiciel devient le cœur du système informatique, apportant de l'intelligence à un produit banalisé (...) ; [il] se positionne de plus en plus comme le produit stratégique dans les systèmes d'information au détriment du matériel" (Eurostaf, 1996 C, p.5 et 13).

³¹ Comme le note Gérard Dréan, "le logiciel est au sens propre l'âme du matériel, c'est à dire ce qui l'anime. Un ordinateur peut être défini comme "une machine à exécuter des programmes", et le matériel n'est qu'un corps vide inerte capable de recevoir une grande variété d'âmes sous forme de programmes" (1996 A, p. 195).

L'évolution est également sensible sur le plan des innovations, Alors qu'il ne s'est pas produit d'innovations majeures dans le matériel depuis vingt ans (Christian Genthon, 1995, p. 93)³², mais essentiellement la poursuite des tendances passées (miniaturisation, augmentation des capacités de traitement, de mémorisation et baisse des coûts), les innovations dans le domaine du logiciel deviennent de plus en plus déterminantes dans l'évolution de l'informatique actuelle et future (cf. chapitre III), même si elles sont fréquemment sous-estimées³³. Déjà en 1985, Christine Afriat et Pierre Caspar relevaient que même les fabricants d'ordinateurs consacraient une part de leur budget de recherche plus importante pour les logiciels (55 %) que pour la mise au point des matériels (45 %), alors que ces parts étaient respectivement de 35 % et de 65 % en 1981 (1988, p. 66).

Il faut noter également que la poursuite des progrès dans les composants électroniques devient de plus en plus tributaires des perfectionnements en matière de logiciels, que ce soient pour les logiciels présents sur les microprocesseurs (microprogrammes), comme pour les logiciels indispensables pour produire des composants intégrant des millions de transistors. La mise au point du logiciel capable d'exploiter les 350 millions de transistors que l'on prévoit de pouvoir intégrer sur un microprocesseur en 2006 constitue un redoutable défi (Jacques Printz, 1998, p. 49). De plus, de nombreux experts, dont Gordon Moore lui-même, considèrent que la "loi" de Moore (doublement du nombre de transistors présents sur un composant tous les 18 mois) s'arrêterait d'ici une dizaine d'années, en raison de limites physiques : la poursuite de la miniaturisation serait impossible, le plus petit composant de transistor à base de silicium mesurerait à peine 4 à 5 atomes d'épaisseur ce qui empêcherait l'isolation. Les autres solutions envisagées (ordinateurs moléculaires, ordinateurs ADN et ordinateurs quantiques) en sont encore au stade expérimental. Dès lors les limites physiques à la poursuite de l'augmentation de la puissance des microprocesseurs ne pourraient plus être contournées que par les progrès en matière d'architectures et de logiciels (Michel Catinat, 1998, p. 39). Par exemple, des recherches sont en cours au MIT (projet Oxygen) sur les perspectives apportées par la

³² Constat partagé par Gérard Dréan qui affirme : "l'innovation et la création des standards ont échappé aux grands fournisseurs de matériels, qui semblent livrer des combats d'arrière garde plutôt que contribuer aux progrès de l'informatique" (1996 A, p.4).

³³ La sous-estimation de l'innovation dans des activités de type services a été analysée par Faïz Gallouj, qui souligne la "tendance à privilégier les aspects "matériels" des phénomènes d'innovation" et qui l'explique par la "transposition de concepts industrialistes et technologistes" (1994, p. 5).

possibilité de rendre le câblage des puces modifiable par les logiciels, de manière à rendre le microprocesseur configurable en fonction des besoins de chaque application.

Le caractère de plus en plus stratégique des logiciels ne se limite pas au secteur de l'informatique. "Les systèmes logiciels sont devenus quelque chose de vital dans l'exploitation des entreprises de haute technologie aussi bien que des entreprises commerciales, vital aussi pour les organisations gouvernementales et militaires" (Capers Jones, 1989, p. 14). Les logiciels jouent un rôle de plus en plus important, notamment dans l'architecture des systèmes d'informations (Jean-Benoît Zimmermann, 1995, p. 183) : "avec la généralisation des technologies numériques, le logiciel conquiert progressivement une dimension essentielle, en tant qu'outil déterminant de cette mutation puisqu'il apparaît comme le support organisateur des flux multiples d'informations traités par les systèmes électroniques" (Jean-Louis Caccomo, 1996, p. 11). Les investissements en logiciels sont un facteur déterminant des différentiels de productivité, à équipement égal, d'une firme à l'autre (Alain Rallet, 1997, p. 88), ce qui explique que les activités consacrées au logiciel représentent une part croissante des activités de recherche-développement des entreprises pour atteindre un cinquième ou plus de la recherche-développement totale des pays où ces chiffres sont mesurés (OCDE, 1997 B, p. 6). La conception des logiciels a une influence croissante sur le type de produits réalisés et surtout sur la façon de les produire, selon des modalités proches des choix concernant les machines auparavant.

L'importance qualitative des logiciels ne se limite pas à la sphère productive. Elle se retrouve dans des domaines très divers depuis les opérations militaires³⁴ jusqu'à de multiples activités sociales et culturelles : les logiciels "commencent déjà à déterminer la façon dont on apprend, dont on travaille, dont on se distrait, dont on se soigne, dont on consomme et aussi dont on forme son opinion" (Roberto Di Cosmo, Dominique Nora, 1998, p. 17). Les logiciels ont une influence sur nos modes de pensée mais sont également déterminés par ceux-ci (Pascal Petit, 1998, p. 29) : par exemple, une étude menée par des chercheurs de la Open University, l'Université de Bournemouth et l'Université Hertforshire montre comment les plus grandes difficultés des filles anglaises face à l'informatique, alors que globalement leurs

³⁴ Dans ce domaine, "la guerre du golfe a suffisamment montré le poids du soft sur le théâtre d'opérations : guidage des missiles, logistique générale, traitement d'images satellitaires, simulation d'environnements" (Daniel Weissberg, 1992, p.73).

résultats scolaires dépassent ceux des garçons, s'expliquent par les "images à caractère masculin" utilisées dans les logiciels (Les News 10/5/99).

Au terme de cette analyse, il apparaît nettement que l'information et les connaissances jouent un rôle croissant dans l'économie, que ce soit au niveau de l'importance prise par les biens et services informationnels, ou au niveau du fonctionnement des organisations comme de l'ensemble de l'économie. Les technologies de l'information et de la communication, technologies "invasives", ont joué un rôle majeur dans cette évolution. Par contre elles ne nous semblent pas conduire à l'existence d'un seul nouveau paradigme, même qualifié de socioéconomique, en raison de la diversité des évolutions dont elles sont porteuses et de la pluralité durable des configurations productives, représentant différentes cohérences possibles entre changements socio-organisationnels, changements techniques et changements économiques.

Le deuxième enseignement majeur concerne le rôle des logiciels au sein des technologies de l'information et de la communication. Malgré une sous-estimation durable de leur importance, et en dépit des difficultés d'évaluation statistique, le rôle croissant des logiciels nous semble un phénomène incontestable et appelé à se poursuivre, ce que nous avons désigné par la "logicialisation" de l'activité économique et sociale.

Chapitre V

LES QUESTIONS CRITIQUES DE L'ECONOMIE DU LOGICIEL

L'économie du logiciel est l'objet depuis ses origines de trois critiques récurrentes : un manque de fiabilité des logiciels produits (résultant soit d'une erreur de comportement du programme dans une situation donnée, soit d'une situation non prévue), une relative inadaptation aux besoins des utilisateurs, et une évolution jugée insuffisante de la productivité dans la production des logiciels avec ses conséquences en terme de coûts et de délais. Les deux premiers aspects expliquent l'importance démesurée prise par la maintenance, que ce soit une maintenance corrective destinée à éliminer les imperfections constatées ou une maintenance adaptative pour améliorer l'adéquation du produit aux attentes des utilisateurs, qui à son tour influe négativement sur la productivité globale du secteur. Vu l'importance prise par les logiciels dans l'économie, "la productivité et la qualité de la programmation (...) sont maintenant des problèmes susceptibles d'affecter les profits et les aptitudes compétitives des entreprises, les délais de livraison des nouveaux produits, la façon dont sont organisées et gérées les entreprises" (Capers Jones, 1989, p. 14).

La réalité de ces problèmes est attestée par la perception des utilisateurs et par l'importance qu'ils occupent dans la littérature spécialisée, qui fourmille d'exemples spectaculaires. Une des explications, fréquemment avancée, est la complexité de l'activité de production de logiciels : "la programmation est ressentie subjectivement comme l'une des plus complexes parmi les activités humaines, et le degré élevé de complexité de la programmation est mentionné comme une raison majeure de sa faible productivité, de ses longs délais et de sa qualité médiocre" (Capers Jones, 1989, p. 91). Les spécificités technico-économiques des logiciels (cf. chapitre III) peuvent également contribuer à expliquer ces problèmes. La résolution de ces problèmes est au cœur du génie logiciel, dont Frederick P. Brooks résume les objectifs : "comment concevoir et construire un ensemble de programmes pour en faire un système ; comment concevoir et construire un programme ou un système pour en faire un produit robuste, testé, documenté et supporté ; comment garder sous son contrôle intellectuel la complexité à larges doses" (1996, p. 249).

Toutefois au-delà du sentiment généralisé d'une production peu efficace, de produits peu fiables et inadaptés¹, l'analyse précise de l'ampleur réelle de ces phénomènes reste à effectuer. En particulier, la persistance de ces problèmes ne signifie pas qu'il n'existe pas des progrès réels dans la production des logiciels, mais plus vraisemblablement que ces progrès restent insuffisants par rapport à l'augmentation de la quantité et à la diversité des logiciels à produire, aux exigences croissantes des utilisateurs, et à la complexité grandissante de ce qui est informatisé. Il importe également de noter que les évolutions à l'œuvre dans l'économie du logiciel sont plurielles, et que, dans certains cas, des améliorations significatives par rapport à une dimension des problèmes (par exemple la fiabilité) peuvent se traduire par une dégradation par rapport à une autre dimension (par exemple la productivité).

Nous examinerons successivement le problème de la productivité dans la production des logiciels (section I), le problème de la fiabilité des logiciels (section II) et le problème de l'adaptation aux besoins des utilisateurs (section III), en analysant les progrès réalisés et les écarts permanents par rapport à l'évolution des besoins, et en soulignant la pluralité des arbitrages réalisés pour privilégier la résolution de certains de ces problèmes et tenter de sortir de "la crise du logiciel". La présence, peut-être excessive, de constats empiriques et d'exemples concrets, se justifie par l'importance des problèmes pratiques, auxquels, à notre avis, l'observateur doit prêter une attention suffisante pour véritablement "comprendre" ce secteur.

¹ "L'avis unanime des gestionnaires et des responsables industriels, des dirigeants des administrations et des chefs militaires est que la programmation est aujourd'hui la moins professionnelle et la plus gênante de toutes les avancées technologiques. La programmation est universellement jugée trop coûteuse, génératrice de trop d'erreurs et beaucoup trop lente. On estime également qu'elle échappe très largement aux mesures, qu'elle est peut-être impossible à mesurer, et que les prévisions y sont inacceptablement imprécises en matière de fiabilité, de consommation de ressources ou de délais" (Capers Jones, 1989, p. 14-15).

Section I - Une amélioration jugée insuffisante de la "productivité" dans la production des logiciels face à l'évolution de la demande

Il s'agit ici de la productivité dans le processus de production des logiciels qu'il faut distinguer de la productivité apportée par l'utilisation des logiciels ou productivité indirecte (cf. chapitre III). Le constat de Jacques Printz, selon lequel "la productivité de la programmation est le problème numéro un du bon usage des ordinateurs" (1998, p. 322), est très largement partagé². En effet, la productivité dans la production de logiciels est réputée se situer à un niveau désespérément bas et croître très lentement (quand elle n'est pas considérée comme étant stagnante). Existe également un accord assez général sur les causes de cette faiblesse de la productivité : part trop faible de produits logiciels fabriqués en série ; faible réemploi de composants déjà disponibles ; automatisation trop faible, parfois même inexistante, de processus de production pourtant généraux et répétitifs ; intégration perpétuelle de nouvelles techniques sur lesquelles l'état des connaissances n'en est parfois qu'aux balbutiements ; complexité grandissante (pas toujours justifiée) des systèmes fabriqués ; utilisation d'un environnement de travail trop artisanal et individualiste ; manque de rigueur de la part de l'ensemble des participants au cycle de développement dans la définition du service et des fonctions des objets à fabriquer ; cycles de décisions trop lents dans les organisations qui définissent les systèmes (Serge Bouchy, 1994, p. 142-143). Cette faiblesse de la productivité du travail est d'autant plus problématique pour le coût des logiciels produits, qu'elle entretient des tensions sur le marché du travail des informaticiens, ce qui est la cause de leurs rémunérations relativement élevées.

Selon William J. Baumol, l'absence d'amélioration de la productivité dans la production des logiciels justifie le classement de l'informatique dans le secteur "à stagnation asymptotique" (A). Pourtant, malgré les difficultés de mesure de la productivité dans une activité comme la production de logiciels, l'examen des différents indicateurs existants montre que la productivité dans la production des logiciels connaît une croissance significative (B),

² Par exemple, selon Capers Jones, "la productivité de la programmation est devenue un sujet majeur de préoccupations sur le plan international" (1989, p. 14).

même si elle est plus faible que dans le domaine du matériel informatique et si elle reste insuffisante face à la très forte croissance des besoins (C).

A - LA THESE DE WILLIAM BAUMOL : L'ABSENCE D'AMELIORATION DE LA PRODUCTIVITE ET SES CONSEQUENCES

L'examen critique des modèles de William J. Baumol, conçus pour analyser la dynamique de l'ensemble de l'économie, est nécessaire pour notre étude, en raison de l'impact de ces modèles sur les débats concernant l'évolution de la productivité, et surtout parce qu'une des deux activités, qui est étudiée pour appuyer sa thèse est l'informatique et en son sein les problèmes liés à l'importance prise par la production des logiciels.

Le premier modèle de "croissance déséquilibrée" de William J. Baumol (1967) voulait rendre compte des effets des taux de croissance de la productivité plus faible dans les activités de service. L'économie y était divisée en deux secteurs, un secteur "stagnant" et un secteur "progressif". Le secteur "stagnant" comprenait les activités de service dans lesquelles le travail était l'intrant principal et les gains de productivité faibles. Le secteur "progressif" comprenait des activités de production de biens dans lesquelles l'usage croissant de capital et la mise en œuvre de nouvelles technologies entraînaient des accroissements continus du produit par travailleur et, par conséquent des salaires plus élevés. Comme les salaires plus élevés se propageaient du secteur "progressif" vers le salaire "stagnant", les coûts et les prix dans ce secteur devaient croître de manière continue ("maladie des coûts"). De plus la faiblesse de la croissance de la productivité dans le secteur "stagnant" conduisait à prévoir que ce secteur absorberait progressivement des proportions de plus en plus importantes de l'emploi total, qui, par effet de structure, entraînerait un "déclin du taux de croissance général de la productivité dans l'économie".

Vingt années après, William J. Baumol, Sue Anne Batey Blackman, et Edward N. Wolf (1985) présentent un second modèle dans lequel ils ajoutent un troisième secteur "à stagnation asymptotique". Les activités de ce secteur ont une composante très avancée sur le plan technologique (intrants du secteur progressif) et une composante relativement irréductible fortement intensive en main d'œuvre (intrants du secteur stagnant). De ce fait "le dynamisme de ces activités peut se révéler passager et quelque peu illusoire" (p. 816). En effet, ces activités "démarrent comme des activités de pointe dominées par leur composante technologique à forte productivité, mais à mesure que la main d'œuvre représente une part

croissante des coûts globaux (parce que la composante dynamique, novatrice, entraîne une réduction de ses propres coûts), elles finissent par revêtir les caractéristiques des services stagnants" (idem). L'analyse du secteur asymptotiquement stagnant est basée sur des données relatives à deux activités, l'informatique et la télévision. En ce qui concerne l'informatique, la composante progressive est l'équipement informatique, dont "le coût par unité de puissance de traitement semble avoir chuté d'environ 25 % par an" (p. 813). Le segment ayant les caractéristiques du secteur stagnant est la production de logiciels : "entre temps, le coût des logiciels (à fort coefficient de main d'œuvre) représentait une proportion toujours plus grande du coût global d'un système informatique" ; selon William J. Baumol, Sue Anne Batey Blackman, et Edward N. Wolf, la part du logiciel dans le coût d'un système est passée de 5 % en 1973, à 80 % en 1978 et 90 % en 1980 (p. 813), ceci s'expliquant par le fait que "l'élaboration du logiciel demeure une activité essentiellement artisanale et, jusqu'à présent un service stagnant" (idem).

Ces analyses ont trouvé un écho dans les prévisions alarmistes régulièrement effectuées, mais non confirmées jusqu'à aujourd'hui, d'une pénurie structurelle de programmeurs dépassant les simples tensions conjoncturelles sur le marché du travail effectivement constatées. Par exemple, l'OCDE (1991 A) cite une étude américaine, selon laquelle, si les tendances actuelles concernant le matériel et le logiciel se poursuivaient, en 2040, toute la population des Etats-Unis (hommes, femmes et enfants) devrait écrire des logiciels !

B - UNE CRITIQUE PARTIELLE : UNE CROISSANCE DE LA PRODUCTIVITE...

Les modèles de Baumol ont suscité de nombreux débats, concernant notamment le fait qu'ils sont basés uniquement sur l'offre et qu'ils reposent sur certaines hypothèses discutables : prise en compte d'un seul facteur de production, taux de salaire unique dans l'économie, volume de population active constant...(cf. pour une synthèse récente, Mara C. Harvey, 1998). L'hypothèse de base selon laquelle les services ne connaissent pas de gains de productivité ("stagnants" dans le premier modèle, "stagnants" ou "asymptotiquement stagnants" dans le second modèle) a été critiquée par Jean Gadrey. Cette critique est basée sur la remise en question de l'argument central de l'approche de William Baumol : "dans certains cas, le travail est un instrument indirect d'obtention du produit final, alors que dans d'autres cas, pour diverses raisons pratiques, il est lui-même le produit final" (William J. Baumol, 1967, p. 417) ; en effet, il est évident que si l'on considère que pour certaines activités, le travail constitue le produit final, il ne peut y avoir par définition d'amélioration de la

productivité du travail dans ces activités. Or, il est toujours possible de définir le produit d'une activité en le dissociant du travail fourni, même si la mesure de la quantité de "produits" n'est pas toujours simple à réaliser. En appliquant cette méthode pour de nombreux secteurs de services, ce qui permet d'appréhender l'évolution de la productivité à partir d'indicateurs plus adéquats, Jean Gadrey montre empiriquement que "dans aucun cas on ne pouvait parler de stagnation, même approximative, de la productivité au niveau de l'ensemble du secteur" même pour des secteurs "où l'on pouvait observer la progression du poids relatif des activités ayant les plus fortes dimensions relationnelles et professionnelles interactives" (1996 A, p. 218-219).

De même, il est possible de montrer que contrairement à ce qu'affirme William J. Baumol il existe des gains de productivité dans la production des logiciels. Le raisonnement élémentaire suivant en fournit une confirmation intuitive : certes, les effectifs employés pour la production de logiciels connaissent une croissance importante, mais qui reste sans commune mesure avec l'augmentation du nombre et de la puissance des matériels utilisés, alors même que ces matériels ont des utilisations de plus en plus diverses et requièrent donc de plus en plus de logiciels différents. Il en découle logiquement que la productivité du travail dans la production des logiciels a augmenté ce que nous allons tenter de mettre en évidence à partir de l'examen de certains indicateurs empiriques.

Les difficultés pour mesurer la productivité

L'étude de la productivité du travail pour la production des logiciels présente des difficultés. La productivité du travail se définit comme le rapport entre la quantité produite pendant une certaine période (en général l'année) et la quantité de travail nécessaire pour fournir cette production. La mesure du dénominateur (la quantité de travail) nécessite d'effectuer certains choix : l'unité de mesure du travail, qui peut être les effectifs moyens sur la période considérée (ce qui nécessite de raisonner en équivalent temps plein pour tenir compte du travail à temps partiel) ou le nombre d'heures travaillées (productivité horaire) ; la délimitation des travailleurs qui contribuent à la production effectuée avec notamment les problèmes de prise en compte des travailleurs "indirectement productifs", et des choix d'imputation du travail des personnes qui contribuent également à d'autres productions que celles mesurées au numérateur. La mesure du numérateur (la quantité produite, qui nécessite de pouvoir définir une unité de production) pose des problèmes en général plus difficiles à solutionner : son caractère le plus souvent hétérogène nécessite de trouver des indicateurs

pertinents pour pouvoir agréger la production. Ces indicateurs peuvent être des indicateurs techniques (ou physiques) que l'on estime représentatifs de la production réalisée, ou des indicateurs économiques (chiffre d'affaires, valeur ajoutée) ce qui nécessite d'éliminer les évolutions des prix non liés à des changements qualitatifs de la production réalisée. Pour que la notion de productivité ait une certaine pertinence comme mesure de l'efficacité du travail de production dans l'obtention de résultats donnés, il est nécessaire que l'on puisse identifier un output séparable de l'activité économique.

La mesure de l'output en termes réels ne pose pas de problèmes particuliers en cas de "validité de l'hypothèse de nomenclature qui veut que la liste des biens et des services soit plus ou moins une donnée de connaissance commune, c'est à dire relativement appréhendable, en essence et en quantité, par l'ensemble de la communauté concernée" (Pascal Petit, 1998, p. 348). Suivant la plus ou moins grande hétérogénéité de l'output, on pourra soit mesurer la production directement avec des indicateurs techniques (ou physiques), soit mesurer la production par des indicateurs économiques, la nomenclature des produits constitutifs de l'activité permettant de construire un indice des prix pour dissocier évolution "en volume" et en prix. Pour la plupart des produits industriels, il est possible de procéder ainsi de façon à peu près satisfaisante. Une première difficulté existe pour appréhender l'évolution de la qualité, quand les produits sont complexes (le nombre de caractéristiques pertinentes est élevé) et évoluent rapidement, ce qui pose le problème de la saisie comptable des innovations, en particulier de l'innovation dite non proportionnelle "qui améliore la qualité des produits en augmentant la quantité de caractéristiques d'un input relativement à leur coût" (Bernard Guilhon, 1991, p. 530). Les ordinateurs constituent un bon exemple de ces difficultés (cf. les fortes différences de baisse des prix selon les méthodes utilisées pour mesurer l'effet-qualité). Des difficultés supplémentaires apparaissent pour de nombreuses activités de services "quand aucun objet ne vient fractionner le rapport entre les producteurs et les consommateurs, quand les résultats apparaissent comme des produits conjoints des uns et des autres, ou quand la rétroaction et l'adaptation interdisent de repérer des unités standardisées, des séries de biens ou de services comparables dans le temps" (Jean Gadrey, 1996 A, p. 134). Il importe dans ces situations d'analyser avec soin la pertinence des indicateurs existants ou d'en construire de plus adéquats (la productivité dans ce cas pouvant rarement être appréhendée à partir d'un indicateur unique), pour éviter des contresens majeurs sur les tendances à l'œuvre (cf. de multiples exemples dans les travaux de Jean Gadrey).

Le secteur de la production des logiciels combine ces difficultés. En effet, il comprend d'une part des progiciels, produits standardisés et reproductibles, mais qui changent très rapidement et dont la qualité ne peut être appréhendée par la connaissance de quelques caractéristiques (cf. les difficultés pour appréhender la valeur des logiciels, chapitre III). Il comprend, d'autre part, des logiciels sur mesure, qui ont en commun avec certains services d'être "des produits marqués par les aléas de la création, de l'initiative, des relations sociales rétroactives" (Jean Gadrey, 1996 A, p. 76). De plus les produits ou résultats des activités de création de logiciels peuvent être repérés selon des perspectives différentes et selon des horizons temporels plus ou moins longs (cf. la distinction entre caractéristiques d'usages directes et indirectes, analysée dans le chapitre I).

Sans nier ces difficultés et sans prétendre avoir réussi à mesurer "la" croissance de la productivité dans ce secteur, il est toutefois possible à partir de l'examen de certains indicateurs techniques (1) et économiques (2) dont on examinera les limites, de dégager des tendances d'évolution significatives.

2 - La croissance de la productivité mesurée par des indicateurs techniques

a - Le nombre de lignes de codes par personne

La première possibilité consiste à mesurer la productivité à partir d'indicateurs techniques (parfois également appelée productivité physique). En effet, techniquement il existe un output séparable de l'activité (le logiciel), même, si de plus en plus, il ne recouvre qu'une partie de celle-ci (services liés). Le problème est que les logiciels sont très différents les uns des autres et qu'il faut leur trouver une caractéristique commune, pour disposer d'une unité de mesure de ce que produit le processus de développement de logiciels. La plus évidente est qu'ils sont tous constitués d'un code source et que l'on peut facilement connaître le nombre de lignes de codes. Il suffit ensuite de calculer le nombre de lignes de codes sources (LS ou KLS pour Kilo Lignes Source) par homme année (KLS/HA) ou par homme mois (KLS/HM)³ pour disposer d'un indicateur de productivité.

³ A notre connaissance, il n'existe pas de tentative de mesure de la productivité horaire, notamment parce qu'il est souvent difficile de connaître la durée réelle du travail des programmeurs ; Capers Jones signale que la pratique répandue des heures supplémentaires non payées peut avoir pour conséquence de sous-estimer les gains de productivité réels, "un gain de 20 % ou 25 % [risquant] de se traduire uniquement par une diminution des heures supplémentaires non payées" (Capers Jones, 1989, p. 272).

Cet indicateur de productivité ne posait pas trop de problèmes d'interprétation quand la production de logiciels était une activité essentiellement individuelle, basée sur le même langage de programmation et que l'indicateur était appliqué à des programmes d'une taille et d'une complexité comparables. La facilité et le coût très faible d'obtention de cet indicateur fait qu'il est resté "l'unité de mesure la plus répandue dans l'industrie de la programmation" (Capers Jones, 1989, p. 24). Il correspondait à "l'hypothèse de loin la plus répandue au sujet de la productivité, depuis les débuts de l'industrie de la programmation, [qui] était qu'améliorer la productivité voulait dire augmenter la possibilité de rédiger des lignes de code-source à plus grande vitesse" (idem, p. 21).

Le premier constat mis en évidence par l'utilisation de cet indicateur est l'importance de la différence de productivité à qualité constante selon les programmeurs : Jacques Printz (1998, p. 88) a constaté des variations de productivité dans un rapport de 1 à 10, Capers Jones, 1989, p. 128) cite différentes études mettant en évidence un rapport de 1 à 20. A partir de la construction d'un modèle ("*software productivity, quality and reliability model*") utilisé pour normaliser et synthétiser des projets réels, Capers Jones a estimé l'influence de la seule expérience sur la productivité : pour réaliser un programme de 1000 lignes en Cobol, la productivité passe de 100 lignes par mois en moyenne pour un programmeur inexpérimenté à 250 pour un programmeur expérimenté (idem, p. 130). Parmi les programmeurs expérimentés, les différences de productivité peuvent également être importantes : Frederick P. Brooks, (1996, p. 24) cite une étude qui montrait, à l'intérieur d'un groupe de programmeurs expérimentés, que le rapport entre les meilleures et les plus mauvaises performances étaient en moyenne de 10 pour la productivité, et de 5 pour la vitesse du programme et la mémoire occupée. Toran DeMarco et Timothy Lister organisent des *Coding War Games* depuis 1977, compétition (publique depuis 1984) entre informaticiens portant sur la conception, le codage et la vérification d'un programme de moyenne importance ; de 1984 à 1986 ces compétitions ont opposé plus de 600 programmeurs représentant 92 compagnies ; les principaux résultats sont que les meilleurs font mieux que les moins bons dans un rapport de 10 pour 1, le meilleur exécutant est 2,5 fois supérieur que l'exécutant moyen, la moitié supérieure à la moyenne fait mieux que l'autre moitié dans un rapport dépassant 2 pour 1 (1991, p. 54).

b - L'importance des conventions utilisées

Cependant, le nombre de lignes de code source par personne ne peut être utilisé valablement qu'en précisant clairement les nombreuses conventions adoptées, ce qui est loin

d'être toujours le cas, alors même que ces conventions sont fréquemment différentes selon les études effectuées. Concernant le numérateur, Capers Jones dénombre onze façons différentes, concernant les programmes et les projets, pour compter le nombre de lignes de code source. Les variantes au niveau des programmes concernent principalement la définition de ce qui termine une ligne de code (un retour à la ligne ou un séparateur logique), la prise en compte des définitions de données, des commentaires ou uniquement des instructions exécutables. Elles ont pour conséquence des variations d'au moins cinq à un entre la technique de comptage la plus lâche et la technique de comptage la plus compacte. Les variantes au niveau des projets concernent l'éventuelle prise en compte du code réutilisé, des lignes de codes supprimées (qui peuvent demander beaucoup d'efforts et permettre une amélioration du produit par augmentation de la vitesse de traitement et diminution de l'encombrement en mémoire), du code temporaire de mise au point et du code d'assistance ; selon les choix effectués l'estimation de la productivité peut aller de 333 lignes de code par homme-mois à 7666 lignes de codes par homme-mois (Capers Jones, 1989, p. 31 et 32). Concernant le dénominateur, selon que l'on prend en compte uniquement les personnes qui font du codage et que pendant les périodes où elles effectuent celui-ci, ou qu'au contraire on comptabilise l'ensemble des personnes qui participe au cycle de développement du logiciel depuis l'expression des besoins jusqu'à la maintenance des applications, les écarts de productivité sont encore plus importants : ainsi Capers Jones estime que la productivité apparente contenue dans les lignes de code source par homme-année est de 25 000 pour "le codage seul, mesuré pendant une journée et converti en taux annuel", à 250 pour "les moyens totaux consacrés en un an au logiciel par l'entreprise, y compris projets abandonnés, tous les développements et améliorations et toute la maintenance" (1989, p. 38).

L'importance de ces écarts traduit la place de plus en plus limitée occupée par l'écriture du code inclus dans le logiciel par rapport à l'ensemble des activités de développement d'un logiciel. Par exemple, Jacques Printz (1998, p. 278) précise qu'il existe trois types de code : le code fonctionnel livré qui correspond aux spécifications fonctionnelles de l'application, le code non fonctionnel livré dont seules certaines parties comme les messages d'erreurs ou les aides en ligne sont visibles de l'utilisateur, et le code non fonctionnel non livré qui a été écrit pour atteindre le but final et qui correspond à des constructions auxiliaires : générateurs de programmes, maquettes, simulateurs, programmes de substitution... Le plus souvent les estimations ne portent que sur le code livré, ce qui est d'autant plus dommageable que le code non-fonctionnel n'est pas proportionnel au code fonctionnel. Surtout une amélioration

apparente de la productivité qui consisterait à réduire le code non fonctionnel risque de se traduire par une chute de la productivité de la maintenance, le code non fonctionnel représentant "la seule vraie garantie de pérennité des investissements réalisés : (...) le risque [étant] moins grand de voir disparaître le produit, que de voir disparaître ceux qui l'ont fait naître" (idem, p. 281). La détection des erreurs dans un logiciel nécessite également une écriture importante de code non fonctionnel non livré : Frederick P. Brooks (1996, p. 125 et 214) estime qu'il est rentable d'écrire jusqu'à 50 % de code supplémentaire par rapport au produit en cours de débogage dans des "échaffaudages", c'est à dire "des programmes et des données élaborés pour les besoins du débogage, mais qui ne feront pas partie du produit final".

Un deuxième exemple expliquant les écarts selon les activités prises en compte concerne la documentation. Son importance est tout à fait décisive pour l'utilisation et la maintenance des applications réalisées, et elle peut mobiliser des moyens très importants. Capers Jones (1989, p. 39-40) cite l'exemple d'un important système de télécommunication qui était accompagné de 100 différents modèles de documents représentant 60 000 pages et 30 millions de mots. La moyenne de ce projet était de 120 mots anglais par ligne de code-source ce qui n'est pas considéré comme anormalement élevé, les grands systèmes des administrations civiles et militaires comportant 200 mots anglais en moyenne de documentation pour chaque ligne de code du système livré au client.

Enfin indépendamment de ces problèmes qui peuvent se résoudre par une normalisation des conventions utilisées, il faut également tenir compte du "style" de programmation qui peut aboutir à des programmes de taille sensiblement différente pour résoudre le même problème. Un exemple célèbre est le conflit opposant Microsoft à IBM lors de leur projet d'écrire en commun un système d'exploitation (OS2) : la contribution de chaque entreprise, de culture très différente, au projet commun était mesurée par le nombre de lignes de code source produit, et les programmeurs de Microsoft considéraient que les programmeurs d'IBM avaient besoin d'écrire (inutilement) beaucoup plus de lignes de codes qu'eux pour réaliser les mêmes fonctions (Paul Carroll, 1994).

c - Le problème de l'hétérogénéité des langages de programmation

Une complication supplémentaire dans l'utilisation de l'indicateur du nombre de lignes de codes par personne pour mesurer la productivité provient de l'utilisation de plus de 500 langages de programmation différents, ce qui ne permet pas d'effectuer une comparaison

directe pour des programmes écrits dans des langages différents. La création de langages de plus haut niveau avait notamment pour objectif d'améliorer la productivité des programmeurs. Or on constate invariablement que lorsque l'on utilise un langage de plus haut niveau, la productivité mesurée par le nombre de lignes de code-source diminue. C'est ce que Capers Jones (1989, p. 21) appelle le "paradoxe le plus significatif de toute la profession : les langages de haut niveau tendent à réduire, au lieu d'accélérer, la vitesse de production des lignes de codes". L'explication de cet apparent paradoxe est la suivante : plus le niveau du langage est élevé, moins il y a de code à rédiger pour réaliser une même fonction. Certes, le codage d'une ligne dans un langage de plus haut niveau ne prend pas plus de temps (il aurait même plutôt tendance à diminuer légèrement lorsque le niveau du langage s'élève), mais la part relative des activités indépendantes des langages de programmation (spécifications, conception, documentation...) va augmenter, ce qui dégrade apparemment la productivité mesurée par le nombre de lignes de code-source.

L'exemple suivant de trois programmes fonctionnellement identiques rédigés dans trois langages de niveau croissant permet d'illustrer ce raisonnement :

Tableau XXVII
Le paradoxe des lignes de code :
un programme identique dans trois langages différents

	Assembleur	PL/1	APL
Lignes source	100 000	25 000	10 000
Activité (hommes-mois) :			
Spécifications	10	10	10
Conception	30	30	30
Codage	115	25	10
Documentation	20	20	20
Intégration/tests	25	15	10
Total (hommes-mois)	200	100	80
Lignes de code source par homme-mois	500	250	125

Source : Capers Jones, 1989, p. 22

Le même programme nécessite 100 000 lignes de code source dans le langage de plus bas niveau (l'assembleur) et seulement 10 000 dans un langage de plus haut niveau comme APL. Il est clair que l'utilisation d'un langage de plus haut niveau a permis d'augmenter la productivité "réelle" puisqu'il fallait 200 hommes-mois pour développer la version en assembleur et seulement 80 pour développer la version en APL. Mais mesurée en lignes de code source par mois, la productivité a apparemment chuté (de 500 à 125 !). Il est intéressant de noter que ce n'est pas l'activité de codage qui explique ce phénomène (le programmeur en Assembleur codait en moyenne 870 lignes par mois, et 1000 lignes par mois en PL/1 ou en APL), mais la baisse de la part des efforts consacré au codage, ce qui est justement une des motivations pour utiliser des langages de plus haut niveau.

Formulé autrement, ce paradoxe signifie qu'il n'est pas justifié de comparer directement deux programmes de même taille écrits dans des langages différents : un programme de 1000 lignes Cobol demande beaucoup plus d'efforts pour établir ses spécifications, sa conception et sa documentation qu'un programme de 1000 lignes Assembleur parce que ses fonctionnalités sont beaucoup plus importantes.

Il faut pourtant remarquer que le plus souvent l'utilisation très répandue de la mesure de l'évolution de la productivité par le nombre de lignes de code par personne, ne tient pas compte des différences des niveaux de langages utilisés, ce qui a certainement contribué à sous-estimer l'augmentation réelle de la productivité dans la production de logiciels.

Une première possibilité pour éliminer ce biais consisterait à évaluer le logiciel produit en remplaçant la mesure des lignes de code-source, par des lignes de code-objet ou octets occupés en mémoire, une ligne de code source en langage de haut niveau se traduisant par beaucoup plus de lignes de code-objet qu'une ligne de code-source en langage de bas niveau (un programme de 1000 lignes Cobol génère un programme exécutable de taille beaucoup plus importante qu'un programme de 1000 lignes assembleur). Toutefois cette solution, si elle est très simple à mettre en pratique, risque d'aboutir à un nouveau paradoxe : l'amélioration continue des compilateurs permet de générer un code-objet de plus en plus compact, ce qui permet d'améliorer les performances (vitesse d'exécution, place occupée en mémoire) mais ferait apparaître une baisse de la productivité mesurée par la taille du code-objet.

La seconde possibilité, retenue par Capers Jones, consiste à convertir les programmes écrits dans les différents langages en "équivalent assembleur" en se basant sur un nombre moyen d'instructions exécutables générées par une instruction source. En reconnaissant que

l'estimation de ces ratios est approximative, Capers Jones propose les conversions suivantes : une instruction source en assembleur génère une instruction exécutable, une instruction source en Cobol ou en Fortran génère en moyenne trois instructions exécutables (et est donc équivalente à trois instructions assembleur), une instruction source en APL génère en moyenne 10 instructions exécutables, une instruction source en Smalltalk (un langage objet) génère 15 instructions exécutables, une instruction dans un langage de tableur génère 50 instructions exécutables... (1989, p. 69). A partir de ce ratio, on peut calculer l'évolution de la productivité mesurée en unités artificielles d'équivalent assembleur. Pour un système remplissant les mêmes fonctions réalisé à vingt ans d'intervalle, Capers Jones donne les estimations suivantes :

Tableau XXVIII
Productivité de systèmes remplissant les mêmes fonctions
à vingt ans d'intervalle

	1964	1984
Langage	Assembleur	Cobol
Lignes source	30 000	10 000
Activité (hommes-mois) :		
Spécifications	10	4
Conception	15	6
Documentation interne	14	4
Documentation utilisateurs	29	11
Codage	99	20
Tests d'intégration	40	11
Correction des défauts	48	31
Gestion	25	7
Total (hommes-mois)	280	94
Lignes de code source par homme-mois	107	106

Source : Capers Jones, 1989, p. 106

On constate que bien qu'aient été pris en compte des améliorations de productivité dans l'ensemble des activités du cycle du développement du logiciel, le changement de langage de programmation fait que la productivité mesurée par le nombre de lignes de code-source (sans tenir compte du changement de langage) reste stable. Par contre, en considérant qu'une ligne

Cobol équivaut à trois lignes en assembleur, la productivité en équivalent assembleur est de 319 lignes par homme-mois en 1984, soit une hausse de près de 200 % en vingt ans⁴, ce qui correspond à un taux de croissance annuel moyen de 5,6 %. Certains auteurs font état de gains de productivité encore plus importants : par exemple, Frederick P. Brooks estime que "la productivité en programmation peut aller jusqu'à quintupler lorsqu'un langage évolué approprié est utilisé" (1996, p. 207).

Toutefois Capers Jones estime que pour les langages dont le niveau dépasse 15 fois la puissance de l'assembleur, tels les langages à base graphique (par exemple Visual Basic où la programmation repose plus sur l'utilisation de menus déroulants ou des commandes par boutons que sur des instructions déclaratives), la mesure par les lignes de codes perd toute signification et ne doit pas être utilisée (1998, p. 20)⁵.

d - L'augmentation de la taille et de la complexité des projets à réaliser

Jusqu'à maintenant nous avons uniquement considéré des projets de taille identique. Or les projets de taille importante demande des efforts qui croissent beaucoup plus rapidement que l'augmentation du nombre de lignes de codes produites. Par exemple, Jean-Marc Geib estime que l'écriture d'un compilateur Pascal ou C nécessitait un effort de 10 années-hommes (HA), qu'il était de 150 HA pour un compilateur ADA, et qu'il s'élevait à 1000 HA pour le logiciel de la navette spatiale (1989, p. 3). Frederick P. Brooks, manager du logiciel Operating System/360 d'IBM évalue à plus de 5000 années-hommes la conception, la construction et la documentation de ce système entre 1963 et 1966 (Frederick P. Brooks, 1996, p. 25). De façon plus générale, il a été estimé que l'effort exprimé en homme-mois (HM) croît de façon exponentielle avec la taille du logiciel à développer, exprimée en milliers de lignes source livrées (KLS)⁶. Ceci signifie que la productivité des programmeurs décroît fortement avec le volume de code à réaliser (Jacques Printz, 1998, p. 270).

⁴ Capers Jones estime qu'elle est de 300 % mais il s'agit vraisemblablement d'une erreur de calcul.

⁵ Un des effets des générateurs automatiques de code est que, s'ils facilitent considérablement la programmation, ils produisent toujours beaucoup plus de lignes de codes qu'un codage manuel.

⁶ Barry Boehm (1981, pp. 81-84) donne les estimations suivantes dans le modèle de coût CoCoMo : $\text{Effort}_{\text{HM}} = 2,4 * (\text{Volume}_{\text{KLS}})^{1,05}$ pour les programmes simples, et $\text{Effort}_{\text{HM}} = 3,6 * (\text{Volume}_{\text{KLS}})^{1,20}$ pour les programmes complexes. Frederick P. Brooks cite d'autres études selon lesquelles l'exposant serait proche de 1,5 (1996, p. 74 et 206).

En réalité, c'est moins la taille que la complexité de l'application qui affecte la productivité : "il est possible d'envisager de très grandes collections de codes comme par exemple une bibliothèque de macros, un magasin de modules réutilisables ou un ensemble de programmes utilitaires, où la taille totale a relativement peu d'effets sur la productivité d'ensemble parce que les modules individuels sont complètement déconnectés les uns des autres et ne font que se côtoyer dans un même catalogue" (Capers Jones, 1989, p. 122). Mais, le plus souvent, la complexité d'un logiciel augmente de façon non linéaire avec sa taille (Frederick P. Brooks, 1996, p. 159). Gérard Dréan estime que pour les activités de rédaction et de test, la productivité va de 60 instructions (testées) par jour-homme pour un programme simple à moins de 2 instructions par jour-homme pour des systèmes complexes" (1996 A, p.199). Cette chute de la productivité pour la production de grandes applications complexes est d'autant plus gênante que la tentation d'augmenter le nombre de personnes qui travaillent sur le projet pour pouvoir l'effectuer dans des délais raisonnables se révèle peu efficace. C'est ce que Frederick P. Brooks, qui coordonnait jusqu'à plus de 1000 personnes participant au développement de l'OS/360 d'IBM, appelle "le mythe du mois-homme" : les mois et les hommes ne sont interchangeables que lorsqu'une tâche peut être divisée entre plusieurs travailleurs sans réclamer de communication entre eux, ce qui est vrai pour la récolte du blé ou la cueillette du coton mais pas pour la programmation (1996, p. 14). Dans ce cas, il faut prendre en compte la formation des travailleurs au but du projet, à sa stratégie globale, à son plan de travail et aux technologies utilisées, ce qui représente un surcroît de travail qui augmente linéairement avec le nombre de travailleurs. Il faut surtout tenir compte de l'effort supplémentaire de communication : si n tâches doivent être séparément coordonnées avec chaque autre tâche, l'effort augmente en $n(n-1)/2$ (idem, p. 15). Dans des situations extrêmes ces activités supplémentaires font plus que compenser l'apport de travailleurs supplémentaires, ce qui est connu sous le nom de "loi de Brooks" : "ajouter des gens à un projet logiciel en retard le retarde encore davantage" (Frederick P. Brooks, 1996, p. 20)⁷.

Toutefois, il existe une possibilité de contourner cette "loi" et d'arriver à une augmentation significative de la productivité, malgré l'augmentation de la taille moyenne des applications développées. Elle correspond au fait que c'est moins la taille que la complexité

⁷ Sur les effets néfastes des tentatives de raccourcir les délais des projets, on peut également mentionner les travaux de Toran Demarco et Timothy Lister qui montrent que la productivité est meilleure quand il n'y a pas d'estimation initiale du temps nécessaire pour développer le projet (1991, p. 38) !

qui importe. Elle consiste à effectuer un important travail préalable au niveau de l'architecture du système pour le décomposer en petits modules qui doivent avoir une indépendance maximale. Joël Aron à partir de l'étude du développement de neuf grands systèmes chez IBM a montré que la productivité variait de 1500 lignes de code par année-homme quand il y avait beaucoup d'interactions entre les programmeurs et les parties du système à 10 000 quand il y en avait très peu (Frederick P. Brooks, 1996, p. 206). Il apparaît ainsi que "la clé de la productivité est l'architecture", une bonne architecture consistant à mettre en place "les interfaces nécessaires à une croissance indépendante des différentes composantes du logiciel envisagé" (Jacques Printz, 1998, p. 271). C'est du reste le domaine de l'architecture des logiciels qui a connu le plus d'innovations dans la période récente (cf. chapitre III). En même temps, plus les activités de conception se développent au détriment du codage (de plus en plus automatisé), moins il semble pertinent de se baser sur le nombre de lignes de codes pour mesurer la production réalisée.

e - Les autres indicateurs techniques

Malgré "l'importance surprenante historiquement accordée aux lignes de codes" (Capers Jones, 1989, p. 45) alors que presque personne ne se satisfait de cette métrique (idem, p. 61), il existe des tentatives pour construire des indicateurs plus significatifs de l'activité réalisée.

La première tentative est celle de Maurice Halstead (1977) de construire des indicateurs de complexité textuelle à partir d'une démarche scientifique (le titre de son ouvrage, très controversé, est *Elements of software science*). L'idée de base est de comptabiliser séparément les instructions fonctionnelles du programme (appelées "opérateurs") et les définitions de données (les constantes et les variables du programme appelées "opérandes"). Ceci lui permet de mesurer le "vocabulaire" n d'un programme (la somme du nombre d'opérateurs et du nombre d'opérandes différents), et la "longueur" N d'un programme (la somme du nombre d'occurrences des opérateurs et des opérandes). A partir de ces deux grandeurs sont calculées le "volume" du programme (égal à $N \log_2(n)$). Ces indicateurs de base ont été ensuite utilisés pour calculer de nouveaux indicateurs (la "difficulté" D , "l'effort" E , le "contenu d'intelligence" I). Le bilan de cette tentative est mitigé : si la distinction opérée entre les aspects fonctions et données des programmes a permis de mettre en évidence l'importance des données, les indicateurs de base "partagent plus ou moins [avec le nombre de lignes de codes] les mêmes problèmes et les mêmes sources d'ambiguïté" (Capers Jones, 1989, p. 90) ; quant aux autres indicateurs (D , E , I), ils sont "purement subjectifs et ne proviennent en aucune

façon de données objectives" (idem, p. 90-91). Finalement, selon Jacques Printz, "personne n'a été capable de démontrer l'avantage de ces mesures par rapport aux mesures plus traditionnelles" (1998, p. 215).

Une autre tentative consiste à tenter de mesurer la complexité structurelle d'un programme. La mesure la plus connue est celle de Thomas McCabe basée sur la représentation d'un programme par un graphe orienté composés de nœuds (blocs séquentiels d'instructions) et d'arcs (les transferts possibles de contrôle entre les blocs). A partir du nombre de nœuds et d'arcs d'un programme, Thomas McCabe (1976) définit la "complexité cyclomatique" d'un programme. Cette méthode, qui connaît un certain succès notamment pour prédire le nombre de défauts d'un programme, a suscité deux critiques. D'une part, la complexité mesurée est celle de la solution qui a été trouvée et qui, selon les compétences des concepteurs, peut être plus ou moins différente de la complexité du problème posé initialement. D'autre part, cette mesure ne tient pas compte de la complexité des données ; or, selon certains auteurs, la complexité d'un programme vient principalement de la complexité des données qu'il doit traiter (Capers Jones, 1989, p. 95).

Une troisième solution de nature plus empirique est la technique dite des "points de fonction", expression peu adéquate pour une technique qui ne s'occupe pas explicitement des fonctions. Cette technique a été développée par A.J. Albrecht en 1979, qui tentait de mesurer l'évolution de la productivité chez IBM pour des programmes écrits dans une grande variété de langage et qui se heurtait au paradoxe exposé précédemment. La mesure des points de fonction, déterminés à partir des caractéristiques d'un projet de logiciel et indépendants du langage de programmation utilisé, correspond aux totaux pondérés et ajustés de cinq éléments : les entrées de l'application, les sorties de l'application, les fichiers logiques associés à l'application, les requêtes pouvant être effectuées vis à vis de l'application, les interfaces entre l'application considérée et d'autres applications externes. Cette méthode conserve un certain degré de subjectivité dans la détermination des coefficients de pondération et par l'introduction d'une plage de variation autorisée de plus ou moins 25 %, en fonction de certains facteurs comme une grande complexité ou un traitement temps réel.

Cette technique a été utilisée au départ par A.J. Albrecht sur un ensemble de 22 projets s'étendant sur cinq ans. Sur cette période de cinq ans et avec cette métrique, A.J. Albrecht et J.E. Gaffney (1983) ont estimé que la productivité avait été multipliée par 3, ce qui représente un taux de croissance annuel moyen de près de 25 %. Il faut noter que cette période a connu

beaucoup d'améliorations des technologies de développement et des langages de programmation. Pour expliquer cette forte croissance de la productivité, A.J. Albrecht mentionne comme facteurs principaux, l'utilisation de la programmation structurée, des langages de haut niveau, du développement *on-line*, et d'une bibliothèque de développement de programmes.

En constatant en 1989, qu'aucune métrique communément admise par tout le monde n'avait réussi à se substituer au nombre de lignes de code, Capers Jones plaidait pour la création d'un indicateur unique ("équivalent logiciel du "produit étalon" de la comptabilité analytique") qui aurait intégré les points de fonction de A.J. Albrecht, la mesure de la complexité de T. McCabe et une mesure de la complexité des données fondée sur les méthodes de conception par l'analyse des données (la méthode la plus connue étant celle de Jackson, de Warnier et de Orr). L'examen des écrits postérieurs de Capers Jones semble indiquer que cette tentative n'a pas abouti et que ceux qui ne se satisfont pas de la métrique du nombre de lignes de codes recourent principalement à la méthode des points de fonctions, dont les aspects les plus subjectifs ont été en partie éliminés par son application à de nombreux projets et par l'importance des programmes de recherche qui lui ont été consacré (Capers Jones, 1998).

f- Une productivité manifestement croissante

Il est difficile d'obtenir des informations statistiques globales et fiables sur les indicateurs techniques permettant d'appréhender la productivité. Tout d'abord, beaucoup d'entreprises n'effectuent pas ces mesures : "les mesures relatives à la programmation restent le maillon le plus faible de toute la science du génie logiciel" (Capers Jones, 1989, p. 19) ; de même, Serge Bouchy (1994, p. 194) indique que si les indicateurs de complexité ont été partiellement instrumentés dans des outils de logimétrie, ceux-ci restent peu utilisés. Ensuite, les entreprises, en général de grande taille⁸, qui effectuent ces mesures communiquent rarement leurs résultats. Les seules sources sur lesquelles on peut s'appuyer sont les écrits de personnes travaillant (ou le plus souvent ayant travaillé) dans ces sociétés (par exemple Frederick P. Brooks à IBM), et des consultants spécialisés dans le développement de la

⁸ Ces mesures peuvent s'avérer coûteuses : Capers Jones estime qu'IBM consacre l'équivalent de 5 % de tous ses coûts de développement à des opérations liées aux mesures (1989, p. 289).

productivité concernant les logiciels (Tom DeMarco et Timothy Lister, Capers Jones, fondateur et président de la société *Software Productivity Research*).

Ce qui frappe à la lecture de ces écrits, c'est que tous mentionnent des estimations statistiques de gains de productivité élevés (voire dans certains cas très élevés, cf. A.J. Abrecht, supra), tout en les considérant comme minimales. On peut l'expliquer par le fait que ces études émanent d'informaticiens dont la référence semble plus être les gains de productivité dans la production de matériel informatique, que dans l'ensemble de l'économie. Par exemple, Jacques Printz cite une étude de ACM SIGSOFT (Vol 8, n° 2 April 1983) qui indiquait que la productivité moyenne d'un programmeur avait été multipliée par 3,6 en 30 ans –ce qui représente un taux de croissance annuel moyen de 4,36 % - et considère qu'il s'agit d'une performance "ridicule comparée à la performance réalisée pour le matériel" (1998, p. 233). Une autre explication possible de cette sous-estimation des progrès constatés, est la volonté de relativiser les annonces, fréquentes dans le domaine des logiciels (et souvent commercialement intéressées), d'une nouvelle solution⁹ ou d'un produit miracle, censé à lui seul, multiplier la productivité par un nombre très élevé (la promesse d'une augmentation de la productivité de 1000 % n'est pas rare !).

Les facteurs d'augmentation de la productivité

La thèse centrale du livre de Tom DeMarco et Timothy Lister "Les hommes de l'ordinateur (Les conditions de la productivité des équipes de projets informatiques)"¹⁰ est que "les difficultés majeures de notre travail ne sont pas tant de nature technologique que de nature sociologique" (1991, p. 14). Ayant constaté lors des compétitions qu'ils organisent régulièrement entre des programmeurs de différentes entreprises (cf. supra), que les écarts de productivité très importants sont moins des écarts entre des individus que des écarts entre les différentes entreprises, ils prodiguent de nombreux conseils sur la gestion des ressources humaines pour atteindre les performances des meilleures sociétés. Cette insistance sur les facteurs humains est justifiée par le fait que les concepteurs de logiciels "travaillent dans la

⁹ "La communauté du génie logiciel se distingue de tous les autres groupes professionnels par le fait qu'elle généralise immédiatement une idée utile et la prend ensuite comme solution par excellence pour chaque problème", avec une "croyance presque religieuse à cette nouvelle pierre du savoir" (Maarten Boasson, 1998, p. 5-6).

¹⁰ Le titre anglais est "Peopleware : productive projects and teams". La traduction de *peopleware* par "les hommes de l'ordinateur" n'est pas des plus judicieuses ; "humanciel" aurait peut-être mieux rendu compte de la pensée des auteurs.

communication et sont au service des utilisateurs pour donner forme à l'expression de leurs besoins. Cette fonction restera primordiale, quels que soient les changements qui interviendront par ailleurs, et elle est peu susceptible d'être automatisée" (idem, p. 43). C'est ce qui explique que, "la productivité dans l'industrie du logiciel a progressé au rythme de 3 à 5 % par an, à peine mieux que dans l'industrie de l'acier ou de l'automobile", pourcentage qu'ils reprennent d'une étude de 1979, et qu'ils estiment "malheureusement très faible" (idem, p. 42).

A partir de la construction d'un modèle basé sur des projets réels, Capers Jones identifie vingt facteurs dont il a pu quantifier l'impact sur la productivité et la qualité de la programmation, et vingt-cinq autres facteurs significatifs "pour lesquels les informations ne sont pas suffisantes pour pouvoir porter un vrai jugement quantitatif" (Capers Jones, 1989, p. 111-112). Outre l'influence des langages de programmation, les principaux facteurs quantifiés qui permettent d'améliorer la productivité, sont l'utilisation des méthodes de programmation structurée, d'outils et de conditions d'environnement favorables, de générateurs d'applications et de programmes, du prototypage rapide... En consolidant quelques-unes de ces variables, Capers Jones estime que la productivité est plus de deux fois supérieure pour une entreprise utilisant des méthodes de pointe par rapport à une entreprise utilisant des méthodes d'arrière-garde (idem, p. 251). En prenant en compte l'ensemble des facteurs, Capers Jones juge qu'en 1985, les programmeurs des entreprises de pointe sont capables de fournir presque dix fois plus de fonctionnalité par unité de travail que les moyennes de l'industrie de 1975 (idem, p. 16). Un facteur d'amélioration de la productivité particulièrement prometteur était la réutilisation de composants déjà développés, mais en 1985, Capers Jones jugeait que c'était "une technologie naissante, encore immature et peu formée" (idem, p. 189) et insistait sur l'importance des investissements à effectuer : la mise au point d'une bibliothèque de schémas et modules normalisés peut demander plusieurs années et coûter plus d'un million de dollars. En 1998, Capers Jones souligne que la réutilisabilité offre la plus grande valeur ajoutée et le meilleur retour sur investissement (Capers Jones, 1998, p. 23). En insistant sur le fait que la réutilisabilité ne se limite pas au code, mais peut concerner douze composants réutilisables (architecture, cahier des charges, plans, estimations, conception et spécification, interfaces, données, écrans et éléments d'écrans, code source, documents d'utilisation, plans et tests, jeux de tests), Capers Jones estime qu'une réutilisation réussie – ce qui exige un contrôle de la qualité qui approche le niveau du zéro défaut - améliore de 65 % la productivité, réduit de 50 % les délais et de 85 % les défauts (1998, p. 35).

"L'essence" et la "substance" des logiciels

Une contribution majeure au débat sur l'évolution de la productivité dans la production de logiciels est un article de Frederick P. Brooks, écrit en 1986¹¹ et intitulé "Pas de balle d'argent : l'essence et la substance en génie logiciel". Cet article suscita de nombreux articles de réfutation, auxquels Frederick P. Brooks répondit en 1996 dans "Pas de balle d'argent. Seconde salve". L'article de 1986 fut souvent interprété comme une démonstration du caractère inéluctable de la faiblesse des gains de productivité dans la production des logiciels, ce que pouvait laisser entendre son titre : les "balles d'argent" sont des armes mythiques capables d'abattre par magie les loups-garous, en l'occurrence les projets logiciels, "monstres crachant des retards, des dépassements de budgets et des produits déficients" (Frederick P. Brooks, 1996, p. 156). En réalité, la thèse défendue par Frederick P. Brooks n'était pas si pessimiste : il affirmait qu'"aucun développement en génie logiciel ne produirait, à lui seul, un gain d'un facteur 10 en productivité de programmation dans les dix ans à venir" (idem, p. 184) et il précisera dix ans plus tard, que "l'article prédisait que *conjointement* les innovations en cours de développement en 1986 engendreraient effectivement un gain d'un ordre de grandeur en productivité", (idem) ce qu'il juge avoir été une prédiction un peu trop optimiste. Ce qui explique que des gains de productivité aussi importants ne soient pas perçus comme tels, c'est que la référence explicite est le fantastique progrès qu'a connu le matériel informatique : "on ne peut entrevoir aucune invention qui augmente la productivité, la fiabilité et la simplicité du logiciel, comme l'électronique, le transistor et l'intégration à grande échelle l'ont fait pour le matériel informatique (...) [avec] des gains d'un facteur deux tous les deux ans" (idem, p. 157) ; "aucune percée technologique ne nous promet les résultats magiques dont nous sommes si familiers dans le domaine du matériel" (idem, p. 170).

Ceci s'explique par l'existence de difficultés spécifiques à la production de logiciels. A partir d'une distinction entre l'essence et la substance d'un phénomène, reprise d'Aristote, Frederick P. Brooks divise les difficultés de la technologie logicielle en "essence" (les difficultés inhérentes à la nature du logiciel) et en "substance" (les difficultés qui gênent la production de logiciel, mais n'y sont pas inhérentes).

¹¹ Nous nous référons à la traduction française publiée dans la deuxième édition du "Mythe du mois-homme. Essais sur le génie logiciel" (1996).

L'essence d'une entité logicielle est un édifice fait de concepts étroitement imbriqués. La partie la plus difficile du développement d'un logiciel consiste à bâtir les structures conceptuelles complexes qui forment l'entité logicielle abstraite. En effet, ces tâches "essentiels" seront toujours difficiles en raison de quatre "propriétés intrinsèques de cette essence irréductible des systèmes logiciels modernes" (idem, p. 158) : la *complexité*, la *conformité* ("le logiciel doit se conformer à ce qui existe parce qu'il est le dernier arrivé sur le terrain" et "parce qu'il est perçu comme étant le plus facile à rendre compatible au reste"), la *variabilité* ("le logiciel d'un système lui fournit sa fonctionnalité, et celle-ci est la plus sujette au changement" et "le logiciel peut être modifié plus facilement que les objets matériels : ce n'est que de l'abstraction, infiniment malléable") et *l'invisibilité* ("un logiciel n'a pas de réalité intrinsèque dans l'espace", "les structures d'un logiciel restent intrinsèquement impossibles à visualiser", et les multiples représentations partielles possibles "ne donnent qu'une vision très réduite de l'enchevêtrement touffu du programme" ce qui "freine la conception chez une personne donnée et limite en outre sévèrement la communication entre personnes différentes").

La substance du logiciel est la représentation des entités abstraites (constitutives de l'essence du logiciel) dans un langage de programmation et leur conversion en langage machine. Les tâches substantielles consistent à effectuer cette représentation et à tester sa fidélité. Dans les premiers temps de la programmation, les tâches substantielles dominaient largement les tâches essentielles. De ce fait, les innovations concernant les tâches substantielles ont permis des "améliorations de plusieurs ordres de grandeur" (idem, p. 186) : les trois principales innovations ont été l'adoption progressive des langages de haut niveau, qui a multiplié la productivité par cinq, la substitution du temps partagé au traitement par lots, dont l'impact est jugé moins important, et l'utilisation largement répandue d'environnements de développement unifiés, qui a doublé ou triplé la productivité, voire plus (idem, p. 161-162).

Mais, du fait de ces améliorations, les tâches substantielles ne représentent plus, en 1986, qu'environ la moitié du total, voire moins, des tâches de développement d'un logiciel¹².

¹² D'une certaine façon on retrouve à l'intérieur même de la production des logiciels, la coexistence, qui est à la base des modèles de Baumol, d'activités ayant des potentiels de gains de productivités différents, les activités dont la croissance de la productivité est la plus forte (ici les tâches "substantielles") voyant logiquement leur importance relative diminuer. Par contre la divergence importante avec les hypothèses de Baumol est que les

Or, à partir du moment où les tâches substantielles constituent moins des neuf dixièmes du total, des innovations qui ne concernent que les tâches substantielles ne peuvent permettre de décupler la productivité, même si elles parvenaient à réduire à zéro les tâches substantielles (idem, p. 181). A partir de ce raisonnement, Frederick P. Brooks en déduit que les innovations des années quatre-vingt dont beaucoup espéraient qu'elles permettraient de résoudre les problèmes de productivité concernant la production de logiciels, ne pourront multiplier la productivité par plus de dix. En effet, selon l'auteur ces innovations "permettent d'améliorer uniquement les composantes d'expression des concepts" (idem, p. 169). Ces innovations, qualifiées de "fausses balles d'argent" sont l'utilisation d'Ada comme langage universel, la programmation à objets, l'utilisation de l'intelligence artificielle et des systèmes experts, la programmation "automatique", la programmation graphique (ou visuelle), la vérification de programme, et de nouvelles améliorations concernant l'environnement de programmation, les outils et les postes de travail.

Les facteurs qui concernent "l'essence" des logiciels

Cependant, en conclusion de son article, Frederick P. Brooks pointe quatre facteurs qui "s'attaquent à l'essence conceptuelle", des "approches très prometteuses", qui prennent pour cible l'essence du problème logiciel, c'est à dire l'élaboration de ces structures conceptuelles complexes. Il s'agit tout d'abord de la détection systématique "des grands concepteurs", en observant que "la construction de logiciel est un processus *créatif*" et que "les logiciels qui ont déchaîné la ferveur d'adeptes passionnés sont ceux qui ont été créés par une poignée de grands concepteurs, voire par un seul" (idem, p. 174).

Le deuxième facteur est le développement d'approches et d'outils de prototypage rapide de systèmes, utilisés pour la spécification itérative du cahier des charges. En effet, selon Frederick P. Brooks, "la plus importante fonction que remplissent les constructeurs de logiciels vis-à-vis de leurs clients est l'extraction et l'affinage progressif des spécifications du produit" (idem, p. 172). Décider exactement ce qu'il faut construire, établir des spécifications techniques détaillées, y compris toutes les interfaces avec les utilisateurs, les machines et les autres systèmes logiciels, sont les parties les plus difficiles de la construction d'un logiciel. Elles peuvent avoir des conséquences catastrophiques sur le système résultant si elles sont mal faites, et elles sont les plus difficiles à rectifier ultérieurement. C'est pourquoi les possibilités

tâches essentielles ne se caractérisent pas par une absence de gains de productivité, même si ceux-ci sont moins élevés que pour les tâches "essentielles" (cf. infra).

d'effectuer de nombreuses itérations entre le client et le concepteur sont la source d'améliorations essentielles.

Le troisième facteur, qui est complémentaire du précédent, est le développement incrémental. Il s'agit par analogie avec l'évolution naturelle de ces structures si complexes que sont les êtres humains, de "faire grandir les logiciels au lieu de les construire" (idem, p. 173). En se basant sur une conception descendante, on commence par faire "tourner" un système embryonnaire, qui ne fait pas grand chose d'utile, puis progressivement on le dote de ses différentes fonctionnalités. Ce mode de développement a une influence très positive sur le moral des développeurs, permet de vérifier en permanence la conformité du système aux attentes des utilisateurs et de revenir facilement en arrière.

Le quatrième facteur est la "réutilisabilité", vu la faiblesse des coûts de reproduction de ce qui a déjà été développé. Celle-ci peut être pratiquée au niveau des différents composants d'un logiciel ou au niveau du produit global (progiciel). Selon Frederick P. Brooks, la réutilisation des composants se heurte à certains obstacles : elle nécessite que chaque composant puisse être spécifié complètement, sans être dépendant des propriétés d'autres composants (Maarten Boasson, 1998, p. 9) ; les composants doivent bénéficier d'une bonne conception et d'une excellente documentation, ce qui les rend beaucoup plus coûteux à réaliser que les composants à usage unique (nécessite trois fois plus d'efforts selon Frederick P. Brooks) ; surtout, il suffit que le consommateur potentiel *pense* (indépendamment du coût réel) qu'il sera plus coûteux de trouver et de vérifier un composant qui satisfasse ses besoins plutôt que d'en écrire un, pour qu'il écrive un nouveau composant, "réinventant la roue". Dans sa réponse de 1996, Frederick P. Brooks estime que si la réutilisation des composants a suscité beaucoup d'études, elle est "relativement peu mise en pratique aux Etats-Unis" à l'exception de quelques "communautés" (les mathématiciens, les développeurs de codes de calculs des réacteurs nucléaires, de modèles climatiques ou de modèles océaniques). Dans de tels cas, ce qui facilite la réutilisation c'est que les composants nécessitent pour leur mise au point un énorme effort intellectuel, et qu'il existe une riche nomenclature normalisée pour décrire les fonctionnalités de chaque composant.

Par contre la production de progiciels apparaît comme offrant des perspectives radicales d'amélioration de la productivité : "le développement du marché de masse est à mon avis la tendance à long terme la plus marquante du génie logiciel" ; en effet, "le coût du logiciel a toujours été un coût de développement pas un coût de duplication. (...) Partager ce coût entre

plusieurs utilisateurs (...) diminue radicalement le coût par utilisateur (...). La mise en service de n exemplaires d'un système logiciel multiplie par n la productivité de ses développeurs", (Frederick P. Brooks, 1996, p. 171)¹³. L'utilisation des progiciels a suscité au départ certaines réticences, en raison notamment d'exigences trop spécifiques des utilisateurs, mais la baisse du coût du matériel a favorisé leur diffusion¹⁴ à des centaines de milliers (et dans certains cas à des millions) d'utilisateurs.

Cependant dans le cas des progiciels on peut de moins en moins mesurer l'évolution de la productivité en se basant uniquement sur les efforts de développement du logiciel original. Si les coûts de reproduction sont dérisoires, il faut également prendre en compte les activités de marketing, de commercialisation, et les services de support liés dont la part relative augmente fortement. De ce fait, l'utilisation d'indicateurs techniques de productivité, qui dans les hypothèses les plus favorables permettent uniquement d'évaluer l'évolution de l'effort de développement, devient de plus en plus inadaptée, et il est nécessaire de recourir à des indicateurs économiques de productivité. Toutefois il importe de préciser que pour toutes les sociétés qui produisent des logiciels, dont certaines sont appelées –improprement – éditeurs de progiciels, l'essentiel de la production consiste à développer et à mettre au point des logiciels. Même chez Microsoft, éditeur de progiciel qui supporte les dépenses les plus importantes de marketing, de commercialisation et de communication, la majeure partie des salariés est affectée aux activités de développement des logiciels.

3 - La croissance de la productivité mesurée par des indicateurs économiques

La mesure "traditionnelle" de la productivité, notamment dans les comptes nationaux, consiste à mesurer la production par la valeur ajoutée, à pratiquer une double déflation pour éliminer les évolutions de prix et à la diviser par les effectifs (ou les heures de travail). Les études qui traitent de la productivité pour la production des logiciels ne procèdent pas ainsi :

¹³ D'une certaine façon, dans le cas extrême mais qui se développe rapidement où un logiciel est disponible sur un serveur et où le nombre de fois où il est téléchargé par des utilisateurs n'entraîne aucun coût supplémentaire pour le prestataire, se manifeste une forme "d'effet d'audience" (Jean Gadrey, 1996 A, p. 215). Cet effet vise à montrer comment dans des prestations "en public", dont l'exemple typique est le quintette à vents mis en exergue par William Baumol, on peut observer des gains de productivité, à condition de "se placer du point de vue de la consommation des prestations, et non du point de vue de leur production".

¹⁴ "En 1960, l'acheteur d'une machine de deux millions de dollars pouvait se permettre d'en dépenser 250 000 de plus pour un programme de paie personnalisé (...). De nos jours, les acheteurs de machines de gestion à 50 000 dollars ne peuvent absolument pas se permettre de faire leurs propres programmes ; ce sont leurs procédures de paie qui s'adaptent aux progiciels disponibles" (Frederick P. Brooks, 1996, p. 171).

tout d'abord, la plupart mesurent la production par le chiffre d'affaires, ce qui ne pourrait permettre que d'évaluer ce que André Vincent (1968) appelle la productivité brute, moins significative que la productivité nette construite à partir de la valeur ajoutée. Surtout ces études utilisent des indicateurs non déflatés (a), ce qui s'explique par les difficultés pour déflater les données, en l'absence d'indice des prix adéquats (b).

a - L'utilisation d'indicateurs non déflatés

Ces indicateurs sont le chiffre d'affaires par employé, et plus rarement la valeur ajoutée par employé, mesurés en monnaie courante. S'il peut paraître abusif de parler de productivité à propos de ces indicateurs –ce que font le plus souvent les études citées - leur connaissance fournit néanmoins quelques informations intéressantes sur le secteur de la production de logiciels.

Eurostaf fournit des données sur le chiffre d'affaires et les effectifs de la plupart des entreprises produisant des logiciels. Le calcul du chiffre d'affaires par employé pour quelques producteurs importants au niveau mondial montre l'importance des écarts entre les sociétés.

Tableau XXIX
Chiffre d'affaires par employé de quelques producteurs importants de logiciels
(en 1993)

Données 1993	CA (millions de \$)	Effectifs	C.A. / Effectif (en milliers de \$)
Microsoft	4649	14430	322,18
Computer Sciences	2583	28600	90,31
Computer Associates	2148	7200	298,33
Cap Gemini Sogetti	2076	20559	100,98
Novell	1123	4429	253,56
Adobe	313	999	313,31

Source : calculs effectués d'après les données Eurostaf (1995 B, p. 63).

L'examen de ce même indicateur à quatre années d'intervalle (en monnaie courante) pour les principales sociétés européennes montre qu'il existe également des différences importantes dans l'évolution du chiffre d'affaires par employé, particulièrement pour les entreprises produisant principalement des progiciels.

Tableau XXX
Evolution du chiffre d'affaires par employé des principales sociétés européennes
(C.A. par employé en milliers de F. courants)

	1991	1995	Taux de croissance	TCAM
Axime	636,62	739,08	16,09%	3,80%
SG2	509,1	596,31	17,13%	4,03%
Sligos	592,38	635,6	7,30%	1,78%
<i>Baan</i>	540,1	706,48	30,81%	6,94%
<i>CCMX</i>	671,96	634,78	-5,53%	-1,41%
<i>Misys</i>	486,96	523,7	7,54%	1,84%
<i>Multihouse</i>	792,94	704,63	-11,14%	-2,91%
<i>SAP</i>	983,34	1185,62	20,57%	4,79%
<i>Software AG</i>	850,6	834,73	-1,87%	-0,47%
<i>Sopra</i>	553,05	512,95	-7,25%	-1,86%
BSO/Beheer	478,37	482,72	0,91%	0,23%
Cisi	444,27	437,25	-1,58%	-0,40%
Data Sciences	529,25	505,42	-4,50%	-1,15%
debis Systemhaus	* 906,1	1014,3	11,94%	5,80%
Finsiel	527,1	629,59	19,44%	4,54%
Getronics	947,92	1074,23	13,32%	3,18%
Logica	426,43	503,11	17,98%	4,22%
Steria	485,39	494,30	1,84%	0,46%
Syseca	521,43	586,67	12,51%	2,99%
Unilog	472,33	487,25	3,16%	0,78%
Cap Gemini	558,01	544,62	-2,40%	-0,61%
Computer Sciences	378,67	442,94	16,97%	4,00%
EDS	533,01	612,63	14,94%	3,54%
Sema Group	435,79	528,17	21,20%	4,92%
Centre Inf. Catalogne	280,66	303,46	8,12%	1,97%
Centrisa	371,77	334,62	-9,99%	-2,60%
Datev	703,11	726,52	3,33%	0,82%

* 1993

En italique, les sociétés produisant principalement des progiciels.

Source : Calculs effectués d'après les données Eurostaf (1996 B, p. 19).

Pour les principales sociétés françaises, où l'on dispose également d'indications sur la valeur ajoutée, on constate les mêmes disparités, avec des évolutions qui peuvent être très contrastées d'une année à l'autre.

Tableau XXXI
Evolution de la valeur ajoutée par employé des principales sociétés françaises
(en milliers de F. courants)

	1992	1993	93/92 (en %)	1994	94/93 (en %)
Cap Gemini	374	370	-1,1	384	+2,7
CCMX	414	351	-15,2	361	+2,9
Cegid	396	440	+11,1	465	+5,7
Sema (K.E.)	35	36	+2,7	43	+19,5
Sligos	377	368	-2,4	357	-3
Sogeris	1087	475	-56,3	615	+29,5
Sopra	394	351	-10,9	363	+3,4
Unilog	384	372	-3,1	390	+4,8
Axime	313	374	+19,5	384	+2,7

Source : Eurostaf (1996 D, p. 41).

Pour l'ensemble des sociétés adhérentes à Syntec Informatique, l'évolution du chiffre d'affaires par employé (en francs courants) est inégale selon les années, mais est globalement positive avec un taux de croissance annuel moyen de 3,6 % sur la période 1988-1996.

Tableau XXXII
Evolution du chiffre d'affaires par employé des SSII françaises
(en milliers de F. courants)

Année	C.A. /Effectif	Taux de croissance
1988	480,8	
1989	513	6,70%
1990	543	5,85%
1991	576,9	6,24%
1992	598,5	3,74%
1993	612,4	2,32%
1994	626,9	2,37%
1995	623,3	-0,57%
1996	637,9	2,34%

Source : calculs effectués d'après les données Syntec Informatique citées in Eurostaf (1997 B, p. 240).

Pour la France, on dispose également des données détaillées de l'INSEE pour le secteur 72 "Activités informatiques", sur le chiffre d'affaires, la valeur ajoutée et les effectifs. Pour les effectifs, on a additionné les effectifs salariés et les effectifs non salariés. Pour les effectifs salariés, nous avons calculé leur "équivalent temps plein" à partir des effectifs à temps partiel, en appliquant un coefficient de travail à temps partiel calculé à partir de l'année 1996, année où nous disposons de données sur le nombre de travailleurs à temps partiel et sur l'effectif salarié moyen. Le calcul du chiffre d'affaires par effectif pour les années 1993¹⁵ à 1997 donne les résultats suivants:

Tableau XXXIII
Evolution du chiffre d'affaires par effectif en France
(en milliers de F. courant)

		1993	1994	1995	1996	1997	TCAM
72	Activités informatiques	633,236	623,945	660,389	682,007	712,135	2,98%
721	Conseil en systèmes informatiques	597,040	571,830	603,232	647,448	700,198	4,06%
722	Réalisation de logiciels	606,202	611,917	611,574	661,364	665,311	2,35%
723	Traitement de données	689,039	677,243	746,422	731,217	806,506	4,01%
724	Activités de banques de données	633,638	960,691	763,834	852,618	916,787	9,67%
725	Entretien et réparation de machines de bureau et de matériel informatique	656,771	631,163	784,006	691,900	683,758	1,01%

Source : calculs effectués d'après des données INSEE in Annuaire statistique de la France (Editions de 1996 à 1999) et Tableaux Economiques de la France 1999-2000 pour les données 1997.

¹⁵ On ne dispose pas pour les années antérieures à 1993 des données selon la nouvelle nomenclature (NAF).

La croissance de la valeur ajoutée par effectifs apparaît plus rapide :

Tableau XXIV
Evolution de la valeur ajoutée par effectif en France
(en milliers de F. courants)

		1993	1994	1995	1996	1997	TCAM
72	Activités informatiques	330,448	336,234	366,451	384,838	16,46%	3,88%
721	Conseil en systèmes informatiques	320,759	327,408	365,150	393,108	22,56%	5,22%
722	Réalisation de logiciels	333,209	351,369	390,434	400,105	20,08%	4,68%
723	Traitement de données	346,294	343,613	356,674	383,492	10,74%	2,58%
724	Activités de banques de données	280,469	326,036	351,450	378,635	35,00%	7,79%
725	Entretien et réparation de machines de bureau et de matériel informatique	295,049	284,856	296,402	276,057	-6,44%	-1,65%

Source : calculs effectués d'après des données INSEE in Annuaire statistique de la France (Editions de 1996 à 1999) et Tableaux Economiques de la France 1999-2000 pour les données 1997.

Pour les Etats-Unis la connaissance du chiffre d'affaires¹⁶ et des effectifs du secteur logiciel et services informatiques nous permet de mettre en évidence un doublement (en dollars courants) du chiffre d'affaires par employé entre 1985 et 1995.

Tableau XXXV
Evolution du secteur logiciel et services informatiques aux Etats-Unis
(en dollars courants)

	1985	1995	Taux de croissance	TCAM
C.A. (en millions de \$)	45132	152213	237%	12,9%
Emploi	637409	1083977	70%	5,5%
CA / Employé (en milliers de \$)	708,054	1404,209	98%	7,1%

Source : calculs effectués d'après les données OCDE (1998 A, p. 18-19).

¹⁶ Le titre du tableau de l'étude de l'OCDE ("Le secteur du logiciel : un profil statistique pour certains pays de l'OCDE") dont sont extraites les données concernant la production s'intitule "Valeur ajoutée". Toutefois il est précisé en note que pour les Etats-Unis, il s'agit des "estimations des recettes des entreprises imposables" et pour le Japon (cf. infra) des ventes !

Sur la période 1990-1995 qui se caractérise par une croissance moins rapide du chiffre d'affaires par employé, nous disposons de statistiques par sous-secteurs qui permettent de mettre en évidence un niveau plus élevé de cet indicateur pour les progiciels (un taux de croissance annuel moyen de 3,26 % à comparer à 1,53 % pour les "services de programmation informatique"), avec vraisemblablement des disparités inter-entreprises plus importantes à l'intérieur du sous-secteur des progiciels (cf. infra).

Tableau XXXVI
C.A. par employé pour les services informatiques aux Etats-Unis
(en milliers de \$ courants)

	1990	1995	Taux de croissance global	Taux de croissance annuel moyen
737 Services de programmation informatique, traitement de données, et autres activités de services informatiques rattachées	114,392	139,658	22,09%	4,07%
7371 Services de programmation informatique	141,366	152,658	7,99%	1,55%
7372 Progiciels	146,480	171,941	17,38%	3,26%
7373 Configuration de systèmes informatiques intégrés	132,472	158,522	19,66%	3,66%
7374 Préparation et traitement de données, services de traitement	90,595	139,597	54,09%	9,03%
7375 Services de récupération d'information	74,361	96,467	29,73%	5,34%
Total 7376 à 7379 *	97,206	104,191	7,19%	1,40%

* 7376 : Services de sous-traitance informatique, 7377 Location et leasing informatiques, 7378 Maintenance et réparation informatique, 7379 Services informatiques n.c.a.

Source : calculs effectués d'après les données de l'OCDE (1998 A, p. 23-24)

Pour le Japon, le chiffre d'affaires par employé (en monnaie courante) croît beaucoup plus rapidement qu'aux Etats-Unis mais reste à un niveau beaucoup plus faible qu'aux Etats-Unis. Ceci peut s'expliquer par la plus faible proportion de progiciels dans les logiciels au Japon et par leur introduction plus tardive.

Tableau XXXVII
Evolution du secteur logiciel et services informatiques au Japon
(en dollars courants)

	1985	1995	Taux de croissance	TCAM
C.A. (en millions de \$)	655	6764	933%	26,3%
Emploi	162010	407396	151%	9,7%
C.A/Employé (en milliers de \$)	4,042	16,603	311%	15,2%

Source : calculs effectués d'après les données OCDE (1998, p. 18-19).

Les différentes études statistiques font donc apparaître une croissance de la "productivité" (basée sur des grandeurs nominales) importante pour la production des logiciels, indépendamment des fortes variations selon les années et selon les entreprises. Il est vraisemblable que l'évolution réelle de la productivité est encore plus favorable, les logiciels à qualité constante se caractérisant par la baisse de leurs prix ou, plus fréquemment par une amélioration des fonctionnalités fournies pour un prix donné. Pour vérifier cette hypothèse et pouvoir estimer la croissance de la productivité, il faudrait pouvoir déflater les données.

b - Les difficultés pour déflater les données

Pour pouvoir appréhender l'évolution en volume d'une production à partir de la connaissance de son évolution en valeur, on déflate cette production par un indice des prix mesurant l'évolution des prix pour le secteur considéré, en tenant compte des effets qualités. Malheureusement de tels indices de prix n'existent pas pour le secteur du logiciel, du moins actuellement¹⁷. En effet, "définir, mesurer et enregistrer au cours du temps des prix, et

¹⁷ L'INSEE mène actuellement des études pour construire un indice des prix des logiciels.

construire ensuite des indices de prix, exige le respect d'une condition essentielle : l'existence d'unités d'output suffisamment standardisées, dont la nature qualitative reste à peu près stable au cours de la période d'étude, ou à la rigueur dont les variations de qualité puissent faire l'objet d'une mesure acceptable lorsqu'on modifie la base statistique des biens pris en compte" (Jean Gadrey, 1996 A, p. 194). Il est clair que cette condition est très difficile à remplir pour les logiciels sur-mesure, et que pour les progiciels la mesure de l'effet-qualité est délicate à réaliser. Dans de telles circonstances, qui sont fréquentes dans les activités de services où il est souvent difficile de définir des unités produites et donc des prix unitaires, différentes méthodes, plus ou moins satisfaisantes sont utilisées (Jean Gadrey, 1996 A, p. 66 et suivantes).

Une première méthode consiste à déflater la valeur produite par le secteur en utilisant l'indice général des prix de l'économie¹⁸. Cette méthode ne peut être utilisée que si l'on suppose que l'évolution des prix du secteur n'est pas très différente de l'évolution du niveau général des prix, ce qui n'est manifestement pas le cas pour les logiciels.

Une deuxième méthode consiste à effectuer la déflation à partir d'un indice des prix dont on dispose pour un secteur complémentaire, correspondant en général à des biens. Elle est par exemple utilisée pour déflater la production des architectes-conseils, qui est effectuée par l'indice du coût de la construction (idem, p. 67). Appliquée au logiciel, elle consisterait à déflater la production de logiciels par l'indice des prix du matériel. Les estimations de l'évolution des prix du matériel divergent selon ce que l'on estime le plus représentatif (ordinateur ou composant électronique) et selon les méthodes utilisées pour tenir compte de l'amélioration de la qualité (traitement par chaînage ou méthode hédonique), mais dans tous les cas, l'importance des baisses de prix du matériel (cf. chapitre IV) ferait apparaître une très forte croissance de la productivité pour la production des logiciels. Même si on peut justifier cette façon de procéder, par les complémentarités entre les deux secteurs (des ordinateurs plus puissants permettent d'utiliser des logiciels aux fonctionnalités plus étendues et donc plus complexes à réaliser), il ne nous semble pas que cette méthode puisse être retenue valablement.

¹⁸ Cette méthode est fréquemment utilisée pour les services d'assurance où on déflate les primes par l'indice général des prix à la consommation (Jean Gadrey, 1996 A, p. 199).

Une troisième méthode consiste à déflater par l'évolution des rémunérations des salariés du secteur¹⁹. Cette méthode est très contestable, notamment pour les logiciels sur mesure, où la tarification de fait en régie est le plus fréquemment pratiquée, conséquence d'une situation où il n'existe pas d'output séparable économiquement de l'activité de production. Dans ce cas l'évaluation de la production réalisée repose principalement sur les salaires versés, et cette méthode peut être inadéquate pour appréhender l'évolution de la productivité. En effet, une augmentation des salaires moyens peut très bien ne pas correspondre à une augmentation des prix à qualité constante, mais traduire une augmentation de la complexité moyenne des prestations réalisées et donc être représentative d'une amélioration d'une qualité des prestations réalisées. Le calcul et l'interprétation des indicateurs pour deux entreprises dont l'activité principale consiste à réaliser des prestations sur mesure peuvent permettre d'illustrer ce raisonnement.

¹⁹ C'est la méthode qui est utilisée pour mesurer la production de l'ingénierie "en volume" qui est déflatée par l'évolution des honoraires des ingénieurs-conseils (Jean Gadrey, 1996 A, p. 82).

Tableau XXXVIII
Evolution de quelques indicateurs
de Cap Gemini Sogeti

	1989	1990	1991	1992	1993	TCAM
C.A. consolidé (en millions de F. courants)	7055	9172	10028	11884	11027	
Variation (en %)		30,00	9,33	18,51	-7,21	11,81
Effectif moyen	12974	16490	17971	21675	20898	
Variation (en %)		27,10	8,98	20,61	-3,58	12,66
C.A consolidé/Effectif moyen (en milliers de F. courants)	543,8	556,2	558,0	548,3	527,7	
Variation (en %)	7,1	2,2	0,3	-1,8	-3,8	-0,75
Frais de personnel/Effectif moyen (en milliers de F. courants)	315,5	316,2	329,0	340,8	332,6	
Variation (en %)	7,9	0,3	4,1	3,6	-2,4	1,33
C.A consolidé/Effectif moyen (déflaté)	543,8	554,97	535,10	507,60	500,57	
Variation (en %)		2,05	-3,58	-5,14	-1,38	-2,05

Source : calculs effectués d'après des données Eurostaf (1994 B, p. 44).

Tableau XXXIX
Evolution de quelques indicateurs
de Sema Group

	1991	1992	1993	1994	1995	TCAM
C.A. consolidé (en millions de £ courantes)	412	417	502	596	678	
Variation (en %)		1,21	20,38	18,73	13,76	13,26
Effectif moyen	7441	6919	7254	8154	8737	
Variation (en %)		-7,02	4,85	12,41	7,15	4,10
C.A consolidé/Effectif moyen (en milliers de £. courantes)	55,37	60,27	69,2	73,09	77,6	
Variation (en %)		8,85	14,82	5,62	6,17	8,80
Frais de personnel/Effectif moyen (en milliers de £. courantes)	28,91	32,28	34,45	34,46	37,26	
Variation (en %)		11,66	6,72	0,03	8,13	6,55
C.A consolidé/Effectif moyen (déflaté)	55,37	53,98	58,07	61,32	60,21	
Variation (en %)		-2,51	7,58	5,59	-1,81	2,12

Source : calculs effectués d'après des données Eurostaf (1996 E, p. 62).

Sur des périodes de cinq ans ces deux entreprises connaissaient des progressions voisines de leur chiffre d'affaires (un taux de croissance annuel moyen de 11,8 % pour Cap Gemini Sogeti et de 13,26 % pour Sema Group). Mais dans la mesure où les effectifs augmentaient beaucoup plus rapidement pour Cap Gemini Sogeti que pour Sema Group, l'évolution du chiffre d'affaires par employé était logiquement très différente. Sur cinq ans, cet indicateur baissait en moyenne de 0,75 % par an pour Cap Gemini Sogeti alors qu'il augmentait fortement pour Sema Group (taux de croissance annuel moyen de 8,8 %). Il peut sembler contestable que cet indicateur basé sur des grandeurs nominales rende correctement compte de l'évolution de la productivité. Mais la déflation par l'évolution des frais de

personnel unitaires de l'entreprise risque d'induire également des conclusions erronées. En appliquant cette technique, l'évolution de la "productivité" apparaît négative pour Cap Gemini Sogeti (-2,05 % en moyenne annuelle) et faible pour Sema Group (2,12 %) dans la mesure où on considère que l'activité "réelle" a connu une croissance plus faible que ce que montrait l'évolution nominale du chiffre d'affaires, qui recouvrait pour partie une augmentation apparente des prix estimée par l'augmentation des salaires versés²⁰. Or il est vraisemblable qu'au début des années quatre-vingt-dix, période de récession passagère de l'activité informatique, l'essentiel de l'augmentation des rémunérations provenait d'un effet de structure d'augmentation de la qualification moyenne des personnels, qui s'expliquait par la complexité croissante des prestations à réaliser. Si cette hypothèse est vérifiée, la croissance de l'activité réelle (et de la productivité) serait plus forte que ce que montrent les données déflatées par les rémunérations. Ce qui donne du crédit à cette hypothèse, c'est la comparaison entre les données pour les deux entreprises. Alors que la différence entre les taux de croissance annuels moyens de la "productivité" en monnaie courante était de près de 10 points, elle n'est plus que de 4 points pour les données déflatées. C'est la conséquence de l'évolution beaucoup plus rapide des rémunérations unitaires à Sema Group qu'à Cap Gemini Sogeti. Mais l'explication probable de cette différence est une amélioration plus importante des compétences utilisées, correspondant à un positionnement rapide de Sema Group sur des prestations plus sophistiquées²¹ (intégration de système, infogérance) (Eurostaf, 1996 E, p. 7). En l'absence d'indications complémentaires, la déflation par les salaires apporte peu de connaissances sur l'évolution de la productivité et peut induire des contresens d'interprétation.

Une tentative de déflation pour le sous-secteur des progiciels

Par contre, si on se limite au sous-secteur des progiciels, on peut tenter d'effectuer une déflation par un indice des prix spécifique relativement représentatif. Une étude effectuée par

²⁰ En théorie si l'ensemble des prestations était uniquement effectué en régie et était constitué uniquement de travail direct, la déflation par l'évolution des rémunérations, conduirait à une "productivité" stationnaire, à l'exception d'écarts pouvant résulter des difficultés d'adaptation des effectifs à l'évolution de l'activité, ce que semble confirmer l'évolution des indicateurs de Cap Gemini Sogeti sur la période. Le risque, souligné par Jean Gadrey dans le cas de nombreuses activités de service, est de mesurer "un pseudo-produit, qui reflète mal l'output qui importe vraiment, économiquement et socialement, à savoir les services rendus" et de l'utiliser pour calculer une "pseudo-productivité qui peut fort bien être stagnante ou décliner, alors que l'importance des services rendus (par heure de travail) progresse" (1996 A, p. 169-170).

²¹ Même, si par définition dans le cas d'une prestation sur mesure, il est difficile de définir des catégories fines de produits indépendamment du prestataire, les services informatiques sont répartis (imparfaitement, avec des nomenclatures différentes dans le temps et selon les sources) dans des sous-secteurs où la complexité de l'activité est en moyenne différente (cf. chapitre IV).

Eurostaf sur un échantillon de 18 entreprises de progiciels, le plus souvent leaders dans leur secteur, évaluait la croissance de la "productivité" du travail (mesurée par le chiffre d'affaires par employé en monnaie courante) à 20,31 % entre 1990 et 1994, soit un taux de croissance annuel moyen de 4,73 %, avec de fortes disparités entre les sociétés : SAP avait sa "productivité" qui augmentait de 53,1 % (TCAM de 11,24 %), Microsoft de 65,6 % (TCAM de 13,44 %), Informix de 75 % (TCAM de 15,02 %), le record étant Nat Systems, spécialisé dans les outils de développement clients-serveurs avec une croissance de 107,5 % (TCAM de 20,02 %). Cette étude précisait que les prix des progiciels étaient en baisse, d'où une augmentation réelle de la productivité plus importante, mais ne fournissait pas d'indications sur l'importance de la baisse des prix (Eurostaf, 1996 C, p. 164).

Pour les Etats-Unis, nous disposons de données sur le chiffre d'affaires et les effectifs pour le sous-secteur 7372 "Progiciels", que nous pouvons essayer de déflater. L'étude des prix des progiciels pose des problèmes : évolution spectaculaire des prix en fonction des stratégies commerciales, tarification différenciée selon les utilisateurs (cf. chapitre III). Des indications partielles semblent témoigner de fortes baisses de prix alors même que les fonctionnalités des progiciels augmentent : par exemple, Eurostaf estime que les prix des progiciels bureautiques ont été divisés par 3 en 1993 avec l'arrivée sur le marché des suites bureautiques (Eurostaf, 1995 B, p. 26) ; selon la Software Publishers Association, le nombre d'unités de logiciels d'application pour ordinateur personnel vendues, pour l'Europe de l'Ouest, avaient augmenté en 1994 par rapport à 1993 de 69 %, mais leur valeur correspondante ne s'était accrue que de 4 %, ce qui correspondait à une baisse du prix moyen de 38,4 % (OCDE 1995 B p. 149). Mais à notre connaissance, le seul indice existant aux Etats-Unis est un indice portant sur les logiciels d'application pour ordinateur personnel basé sur la méthode des modèles correspondants, et qui fait état d'une baisse plus modeste. Si nous l'utilisons pour déflater le chiffre d'affaires des progiciels, en faisant l'hypothèse –discutable- qu'il reflète l'évolution des prix de l'ensemble des progiciels, nous obtenons les résultats suivants :

Tableau XXXX
Productivité brute pour les progiciels aux Etats-Unis

	1990	1991	1992	1993	TCAM
C.A nominal (en millions de \$ courants)	16523	18306	21236	24648	
Variation (en %)		10,8%	16,0%	16,1%	14,26%
Evolution annuelle des prix (en %)		-1,6	-6,5	-0,8	
C.A réel (en millions de \$, aux prix de 1990)	16523	18604	23082	27006	
Variation (en %)		12,59%	24,07%	17,00%	17,79%
Effectifs	112800	124400	130800	144800	
Variation (en %)		10,28%	5,14%	10,70%	8,68%
C.A./Effectif (en milliers de \$ courants)	146,480	147,154	162,354	170,220	
Variation (en %)		0,46%	10,33%	4,85%	5,13%
Productivité (C.A. /Effectif en milliers de \$ aux prix de 1990)	146,480	149,547	176,464	186,506	
Variation (en %)		2,09%	18,00%	5,69%	8,39%

Source : calculs effectués d'après des données OCDE (1998, p. 24,25 et 41).

Indépendamment de l'amplitude des fluctuations (qui reflète en partie celle des prix), la productivité brute a connu une croissance importante (un taux de croissance annuel moyen de plus de 8 %). Il faut toutefois noter que ce qui est mesuré c'est l'évolution de la productivité au sein du sous-secteur des progiciels. La forte hausse constatée peut s'expliquer par la diminution de la quantité de travail pour produire un original (cf. infra la productivité mesurée par des indicateurs techniques) mais également (et peut-être surtout) par le fait qu'un même progiciel est vendu à un nombre croissant d'utilisateurs. Par contre, ce que nous ne pouvons

mesurer (en raison de l'absence d'indice des prix adéquat) et qui représente vraisemblablement la plus grande partie des gains de productivité dans la production des logiciels considérée globalement, *ce sont les conséquences de la substitution de progiciels à des logiciels sur mesure*. La méthode, assez proche de certaines techniques parfois utilisées²², qui consisterait à déflater l'ensemble de la production de la branche des logiciels par un indice des prix des progiciels nous semble trop contestable pour pouvoir être utilisée, dans la mesure où il est très peu vraisemblable que l'évolution des prix des progiciels soit représentative de l'évolution des prix de l'ensemble des logiciels. Toutefois une indication des gains de productivité résultant de la substitution des progiciels aux logiciels sur mesure, est constitué par la formidable baisse des prix qui résulte, pour la résolution d'un problème donné (exploitation d'un ordinateur, paye, comptabilité...), du passage d'une solution basée sur des développements sur-mesure à une solution basée sur des progiciels pour remplir les mêmes fonctionnalités²³ : "un progiciel entier aux riches fonctionnalités coûte moins cher qu'une journée de programmeur avec ses coûts indirects" (Frederick P. Brooks, 1996, p. 244).

C - ...MAIS INSUFFISANTE FACE A L'AUGMENTATION DE LA DEMANDE DE LOGICIELS

Que conclure de l'examen de ces différents indicateurs techniques et économiques ? Même si nous ne prétendons pas avoir réussi à mesurer globalement de façon précise et rigoureuse l'évolution de la productivité dans la production de logiciels, nous pensons avoir rassemblé suffisamment d'indices (au sens policier du terme) partiels et convergents pour pouvoir affirmer qu'il y a eu une incontestable augmentation de la productivité dans la production des logiciels. Il est toutefois vraisemblable que ces gains de productivité sont bien inférieurs aux gains exceptionnels constatés pour le matériel informatique. Cependant

²² Par exemple, pour les services bancaires et financiers, le produit en valeur de cette branche est déflaté par un indice des prix de divers services annexes de gestion (location de coffres, gestion de portefeuilles, placement de titres) plus faciles à repérer mais qui ne représentent qu'une faible partie de l'activité (Jean Gadrey, 1996 A, p. 67).

²³ En réalité les fonctionnalités ne sont jamais rigoureusement identiques : en général le passage à un progiciel permet de fournir une gamme plus large de fonctionnalités ; par contre, étant conçu pour répondre à des besoins standards, un progiciel peut être moins adapté aux spécificités de chaque problème concret qu'un logiciel sur mesure, sauf à entreprendre des développements complémentaires.

"l'anomalie n'est pas la lenteur des progrès du logiciel, mais la rapidité de ceux du matériel"²⁴ (Frederick P. Brooks, 1996, p. 157).

En conséquence de l'importance des progrès de la productivité dans le matériel informatique, la poursuite de la croissance du nombre des ordinateurs, de l'augmentation de leur puissance et de la baisse de leurs prix, fait que l'amélioration de la productivité dans la production des logiciels continuera à constituer un défi permanent, face à "l'explosion de la demande de logiciels" (OCDE, 1991 A). Certes la vision "industrialiste" qui, en mettant au premier plan le matériel informatique, considère que le secteur du logiciel est seulement "tiré" par la croissance du matériel²⁵, doit être relativisée. A l'heure actuelle, les relations entre les deux secteurs sont certainement beaucoup plus équilibrées, et dans de nombreuses situations, c'est la volonté de pouvoir utiliser les possibilités nouvelles offertes par les logiciels les plus récents, qui entraîne un renouvellement accéléré du matériel.

L'écart persistant entre les gains de productivité dans le matériel et dans les logiciels confirme la thèse de William Baumol d'une part grandissante des dépenses en logiciels dans les dépenses informatiques, même si les statistiques disponibles (cf. chapitre IV) font état d'une évolution beaucoup moins rapide que celle qu'il mentionne. Par contre l'augmentation incontestable de la productivité dans la production des logiciels nous conduit à rejeter la classification effectuée par William Baumol du secteur informatique (intégrant les logiciels et les matériels) dans les secteurs asymptotiquement stagnants. Il semblerait plus juste de caractériser le secteur informatique comme étant un *secteur asymptotiquement croissant*²⁶,

²⁴ Frederick P. Brooks estime que "l'explosion de la technologie des ordinateurs qui a connu des progrès sans équivalent dans l'histoire humaine" a permis de "multiplier par au moins mille durant ces vingt dernières années" la productivité dans la fabrication du matériel" (1996, p. 222).

²⁵ Cf. par exemple le raisonnement tenu par Frédéric Georges Roux (1991) : "Les progrès de la technologie alliés aux effets bénéfiques de la concurrence, qui fait pression sur les prix, font que la puissance moyenne de chaque unité centrale installée double environ tous les trois ans. A ce rythme, dans dix ans, l'ordinateur moyen sera dix fois plus puissant qu'aujourd'hui. Comme il y en aura dix fois plus, c'est cent fois plus de MIPS qui seront là. (...) Ces MIPS, cent fois plus, les utilisateurs les consommeront, sinon les constructeurs les garderont en stock ou ne les produiront pas. Or, il faut prendre conscience que si cela arrivait, IBM, DEC, les deux à la fois, feraient faillite. C'est une hypothèse que je ne saurais retenir comme probable. Donc les MIPS seront là. (...) Donc, il y aura de nouvelles applications qui utiliseront toute cette puissance nouvelle disponible".

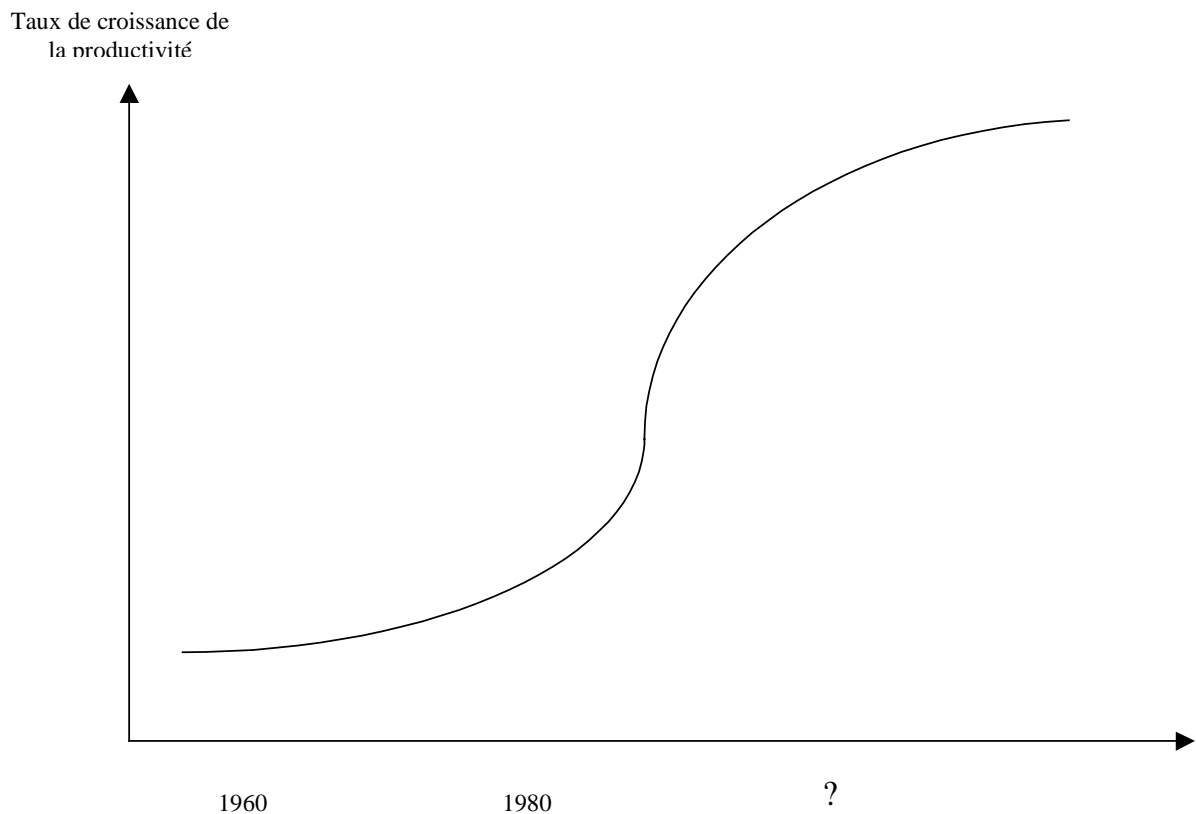
²⁶ William Baumol, au vu d'une version provisoire de cette partie sous la forme d'un document de travail, tout en soulignant son accord avec l'analyse principale et sa conclusion (l'informatique comme un secteur asymptotiquement croissant), estime qu'à long terme "la question de la stagnation peut finalement surgir, la pensée étant un des inputs du développement des logiciels (...) et la pensée moderne n'étant pas plus productive que celle des grands mathématiciens du XVII^{ème} siècle".

dont la pente de l'asymptote est le rythme de la progression de la productivité dans la production des logiciels, tant que la productivité croît plus rapidement dans le domaine du matériel que dans le domaine du logiciel.

La productivité continuera-t-elle à croître plus rapidement pour le matériel que pour les logiciels ? La réponse à cette question est délicate. Du côté du matériel, il semble que la poursuite de la "loi de Moore" (à la base des progrès de productivité) puisse se heurter à des limites physiques d'ici une dizaine d'année, et il est à l'heure actuelle difficile de savoir si de nouvelles technologies (encore au stade expérimental) permettront les mêmes gains de productivité. Du côté de la production des logiciels, où il n'existe pas de limites physiques à la poursuite de la croissance de la productivité, les prévisions sont également très incertaines. Les tendances observées sur les évolutions (passées et actuelles) dans la production des logiciels peuvent toutefois laisser supposer que la croissance de la productivité suit une "courbe en S".

La "courbe en S" de la productivité a été mise en évidence par Jean Gadrey (1999 C, p. 20) pour certains services (commerce, transports, télécommunications, banques et assurances).

Graphique IX



La première phase, correspondant dans la représentation pour les services de Jean Gadrey à un "stade de modernisation lente d'un service traditionnel" (1999 C, p. 20), couvrirait pour les logiciels la période 1960-1980. Dans cette phase, les logiciels sont principalement produits sur-mesure et l'augmentation de la productivité résulte essentiellement des améliorations introduites dans les méthodes de développement, les langages et les outils utilisés.

La deuxième phase qui commence dans les années quatre-vingt, voit la croissance de la productivité s'accélérer, principalement en raison de la part croissante des progiciels dans les logiciels. En effet, la production des progiciels recouvre des activités où les gains de productivité sont moyens (conception de l'original, marketing, commercialisation...) et des activités de reproduction, où l'amélioration de la productivité est très importante avec la croissance du nombre d'exemplaires fournis. Elle comprend également la fourniture de services de support, où les gains de productivité sont plus faibles, mais qui représentent une part modeste de l'activité, au moins dans un premier temps. Elle s'apparente à la phase "de production de masse d'un service standardisé permettant de réaliser des économies d'échelle" (Jean Gadrey, 1999 C, p. 20) observée pour certains services.

Jean Gadrey dans le cas des services identifie une troisième phase dite de "service à valeur ajoutée, où l'on réalise encore des gains de productivité mais à un rythme inférieur" (idem) en soulignant que, dans cette phase, la productivité perd de son sens comme mesure de la performance. Cette évolution correspond à la montée en puissance d'un "sur mesure de masse" dans le cas de la production des logiciels, dont nous avons identifié deux composantes principales : la fourniture avec un progiciel d'une proportion croissante de services divers et variés ; le développement de logiciels sur mesure en réutilisant des modules déjà développés et testés. Dans les deux cas, il s'agit de fournir une solution plus adaptée aux besoins spécifiques de l'utilisateur que ne peut le faire par définition un produit standard (cf. chapitre VII).

Si ces formes de production sont appelées à se développer et si elles sont porteuses d'une moindre croissance de la productivité au sens classique, cela n'implique pas que pour l'ensemble de la production des logiciels, le point d'inflexion de la courbe de croissance de la productivité soit atteint rapidement. En effet, plusieurs facteurs peuvent permettre à la productivité de continuer à croître à un rythme soutenu dans la période actuelle. Tout d'abord les solutions dites "sur mesure de masse" peuvent se substituer à des développements de

logiciels sur mesure (et non à des progiciels) et dans ce cas contribuer à améliorer la productivité. Ensuite et surtout, la production de progiciels pour répondre à des besoins standards a encore des perspectives de croissance très importantes : poursuite de la substitution des progiciels aux logiciels sur mesure dans des domaines encore peu couverts par des progiciels, domaines nouveaux couverts d'emblée par des progiciels, et extension à de nouveaux utilisateurs qui utiliseront des progiciels déjà existants, d'où des gains de productivité très importants.

Par contre, le développement de la fourniture d'une solution adaptée à un problème particulier met au premier plan la nécessité *de prendre en compte d'autres dimensions que la productivité du travail pour apprécier l'efficacité dans la production des logiciels*. Elle repose sur le fait que ce qui importe le plus, est moins la mesure du produit immédiat de l'activité que l'évaluation des résultats indirects de cette activité pour les utilisateurs. Cette évaluation nécessairement multidimensionnelle est encore plus complexe à effectuer dans la mesure où interviennent des facteurs extérieurs à l'activité du prestataire (caractéristiques de l'environnement, compétences des utilisateurs...), où l'horizon temporel à prendre en compte est plus important, et où elle nécessite d'analyser de multiples aspects qualitatifs difficiles à appréhender.

En 1989, Capers Jones notait que "la productivité de la programmation est devenue un sujet majeur de préoccupations sur le plan international, et la qualité des programmes a de fortes chances de prendre bientôt la même importance" (1989, p. 14). La qualité d'une solution logicielle comporte deux dimensions principales : l'adaptation aux besoins des utilisateurs et la fiabilité. Le fait de traiter à part la question de la fiabilité s'explique par son importance particulière pour un produit destiné à réaliser des traitements automatiques²⁷. Elle présente toutefois le risque d'introduire des redondances dans l'analyse, certaines explications étant communes à ces deux caractéristiques fondamentales de la qualité des logiciels.

²⁷ Capers Jones considère même que la fiabilité constitue l'aspect central de la qualité pour un logiciel : "Il existe de nombreuses définitions du mot "qualité" associées à d'aussi nombreuses ambiguïtés. Pour les objectifs pratiques de tous les jours, on peut considérer qu'un système de programmation de qualité supérieure est un système sans un seul défaut susceptible de l'arrêter complètement ou de fournir des résultats inacceptables" (1989, p. 47).

Section II - La fiabilité

La présence de défauts dans les logiciels est certainement l'attribut de la programmation qui affecte le plus les perceptions qu'ont les utilisateurs finals des programmes (Capers Jones, 1989, p. 207)²⁸. En raison de l'importance acquise par les logiciels dans de multiples aspects de la vie économique et sociale, où la sécurité et la fiabilité reposent fréquemment sur des dispositifs logiciels, leurs conséquences peuvent atteindre potentiellement l'ensemble de la population, utilisatrice indirecte et le plus souvent non consciente de multiples composants logiciels, comme l'a révélé la mobilisation impressionnante autour du bogue de l'an 2000. Et pourtant malgré les efforts déployés pour corriger les erreurs, les logiciels contiennent toujours des défauts (A). Ceux-ci étant nécessairement le produit d'erreurs humaines, ils sont de moins en moins acceptés vu le rôle croissant des logiciels (B), alors même que les défauts dans les logiciels sont de plus en plus difficiles à éliminer (C).

A - CONSTAT ET PREMIERES EXPLICATIONS

1 - Les défauts des logiciels

Le constat de la présence de défauts dans les logiciels est très largement partagé par les utilisateurs, la presse spécialisée, qui rapporte de nombreux exemples souvent spectaculaires, et même par les producteurs des logiciels : "les systèmes sont toujours remplis de bogues"²⁹ reconnaissent les dirigeants de diverses grandes sociétés dans des entretiens avec Capers Jones (1989, p. 15). Un défaut dans un logiciel peut, dans certaines circonstances, provoquer une défaillance informatique. Cette défaillance informatique est parfois qualifiée de simple (*fault*) quand on peut la récupérer, ou de grave (*failure*) quand elle provoque l'arrêt définitif du système. Ces défauts concernent tous les types de logiciels, depuis les progiciels destinés au grand public³⁰ jusqu'à des logiciels critiques.

²⁸ Cet auteur souligne que "le taux de défaut est l'élément de coût le plus important de la programmation" (Capers Jones, 1989, p. 16).

²⁹ Un (ou une) bogue est "un défaut de conception ou de réalisation d'un programme se manifestant par des anomalies de fonctionnement" (Journal Officiel du 19 février 1984).

³⁰ Daniel Icbiah (1995), biographe pourtant admiratif de Bill Gates, considère qu'un des plus grands exploits du président de Microsoft, avait été une présentation effectuée pour la commercialisation d'une des premières versions du traitement de texte Word ; alors que ce logiciel contenait encore de multiples défauts, Bill

Le logiciel est à l'origine d'un grand nombre d'erreurs et de pannes des systèmes (OCDE, 1991 A, p. 9). On peut citer, entre autres exemples : le cas d'une personne de 106 ans devant se présenter au CP de l'école communale sous peine de retrait des allocations familiales pour ses parents ; les 22 000 ayants droit de la Caisse d'Allocations Familiales de Vesoul qui ont touché deux fois leurs prestations ; les 486 750 gagnants d'un million de pesos à un jeu promotionnel organisé par Pepsi-Cola à Manille, par suite d'une erreur dans le progiciel devant imprimer le numéro millionnaire ; l'existence de rames de métro "fantômes" à San Francisco ; le cas d'un système d'informations sur la circulation routière affichant que la circulation est nulle quand elle est totalement bloquée ; les premières versions du logiciel de navigation écrit pour le chasseur F16 qui provoquaient un retournement de l'avion lorsqu'il passait l'équateur ; l'inondation en 1983 de la vallée du Colorado River, due à une mauvaise modélisation du temps d'ouverture des barrages. Dans certains cas les explications a posteriori apparaissent très simples (oubli qu'une personne peut dépasser l'âge de 99 ans, fluidité du trafic mesuré par des capteurs se basant sur la vitesse de circulation des véhicules qui n'existaient plus pour le système quand ils ne pouvaient plus avancer du tout), mais certaines défaillances peuvent rester inexplicables très longtemps (le cas des rames de métro "fantômes" n'a jamais été expliqué). Surtout des erreurs minimes peuvent avoir des conséquences majeures : une erreur dans une équation d'un programme (l'omission d'une "barre" sur un symbole indiquant qu'il fallait utiliser des valeurs moyennes) fut la cause de la destruction par erreur d'une fusée Atlas-Agena d'une valeur de 18,5 millions de dollars le 22 juillet 1962 (Lauren Ruth Wiener, 1994, p. 27) ; une virgule à la place d'un point fit passer la sonde de la mission Venus à 500 000 km de la planète au lieu des 5 000 km prévus ; une erreur de signe fit pointer une fusée russe sur Hambourg au lieu du pôle Nord (Patrick Jaulent, 1992, p. 17) ; une variable codée sur 16 bits, alors qu'une autre partie du programme pensait qu'elle était codée sur 32 bits fit perdre cinq millions de dollars à la Bank of New York en 1985³¹. Plus récemment, c'est une erreur de programmation des plus simples qui entraîna la perte d'Ariane V lors de son premier vol en juillet 1996.

Gates avait réussi à faire une démonstration d'utilisation du traitement de texte sans que le système ne se bloque en mémorisant de longues suites d'instructions qui évitaient soigneusement les défauts résiduels.

³¹ Cette variable servait de compteur dans un logiciel pour traiter les transactions sur les titres d'Etat émis par la banque centrale américaine. Ce logiciel, qui fonctionnait normalement depuis un certain temps, devait traiter un nombre de transactions particulièrement élevé le mercredi 20 novembre 1985, et par suite de l'erreur sur la capacité du compteur, se mit à transcrire les nouvelles informations en recouvrant les transactions précédentes ; au bout de 90 minutes quand on constata l'incident et qu'on interrompit les transactions, la Bank of New York devait 32 milliards de dollars à la Réserve Fédérale qu'elle ne pouvait pas récupérer auprès des autres banques (Lauren Ruth Wiener, 1994, p. 33 et 66).

Le nombre réel de défauts que contient un logiciel au moment de sa commercialisation est évidemment inconnu. Une mesure habituelle consiste à recenser le nombre de défauts apparaissant ultérieurement lors de l'exploitation du logiciel, ce qui conduit à une estimation inférieure à la réalité. En effet, certains défauts peuvent très bien ne jamais se manifester ou se manifester après la période d'observation³², ou on ne peut attribuer à un logiciel précis les dysfonctionnements constatés d'un système. Un taux d'erreur est calculé en divisant le nombre d'erreurs découvertes en exploitation divisé par le volume du logiciel en milliers de lignes sources (exprimé en KLS, Kilo Instructions Sources livrées).

L'ordre de grandeur de ce taux d'erreur fait l'objet d'estimations divergentes. Toran Demarco et Timothy Lister considèrent que l'industrie du logiciel a habitué ses clients à accepter des programmes qui présentent en moyenne de dix à trente défauts par milliers de lignes de code (1991, p. 31). Une étude conduite par Capers Jones en 1983 sur de gros systèmes rédigés en assembleur considérait que les logiciels contenaient initialement plus de 50 défauts pour 1000 instructions de code-source ; comme les différentes opérations pour supprimer ces défauts avant la livraison du logiciel (cf. 3) ne permettaient pas d'en éliminer plus de 85 %³³, il restait de l'ordre de 7,5 défauts pour 1000 lignes source dans les programmes livrés aux utilisateurs (Capers Jones, 1989, p. 48). Lors d'une étude ultérieure sur des logiciels commercialisés dont les taux de défaut après livraison ont été quantifiés, Capers Jones estimait que les logiciels contenaient au moment de leur remise au client de l'ordre de 4 ou 5 défauts au moins pour 1000 instructions. Jacques Printz estime que le taux d'erreur se situe entre un et cinq (1998, p. 54). Il s'agit d'estimations moyennes avec des écarts importants. En effet, le nombre de défauts est le "résultat final d'interactions entre la nouveauté du programme ou du système à réaliser, les compétences du personnel chargé de l'étude, les méthodes adoptées pour la conception et le codage, et encore beaucoup d'autres facteurs" (Capers Jones, 1989, p. 211-212). Capers Jones cite le cas d'un programme commercial dont on a découvert 31 erreurs pour 1000 instructions de code source durant les trois premières années d'exploitation (idem, p. 209). Plus récemment le nombre de défauts de Windows 2000 a été estimé à 65 000, à rapporter aux 35 millions de lignes de codes de ce

³² Un défaut fut découvert pour un système de contrôle aérien dans un morceau de programme écrit en Cobol qui n'avait pas été modifié depuis vingt ans (Lauren Ruth Wiener, 1994, p. 137).

³³ Un taux d'élimination des défauts pour l'ensemble des opérations avant livraison de 85 % est selon l'auteur considéré comme maximal. Dans les cas les moins favorables observés par Capers Jones, le rendement d'élimination des défauts n'était que de 44 % (1989, p.209).

logiciel. Par contre la NASA arrive à des taux beaucoup plus faibles : 0,11 erreur/KLS pour le logiciel embarqué de la navette spatiale (qui avait un volume de 500 KLS ce qui fait 55 erreurs)³⁴, et 0,4 erreur/KLS pour le logiciel au sol d'un volume de 1700 KLS, soit 680 erreurs (Jacques Printz, 1998, p. 53).

Il faut toutefois signaler que rapporter le nombre d'erreurs aux lignes de code source présente les mêmes biais statistiques que ceux mis en évidence dans l'évaluation de la productivité, en raison de l'hétérogénéité des langages de programmation. En effet, surtout pour les programmes d'une certaine taille, la majeure partie des erreurs ne sont pas des erreurs de codage, mais des erreurs imputables aux spécifications et à la conception (Capers Jones, 1989, p. 212). C'est pourquoi il est plus pertinent, mais plus coûteux à mesurer, de diviser le nombre de défauts résiduels par le nombre de points de fonction. En 1998, Capers Jones estime qu'en moyenne un logiciel livré contient environ 0,75 défaut résiduel par point de fonction, correspondant à cinq défauts latents par point de fonction, avec une norme d'élimination des défauts de 85 % (Capers Jones, 1998, p. 25).

Les conséquences de ces défauts comprennent les coûts de réparation de ce système, mais aussi les coûts liés à l'arrêt du système qui ne rend pas les services attendus. Ces coûts peuvent être évidemment très variables selon le caractère plus ou moins critique du logiciel³⁵, depuis la simple impossibilité d'effectuer des opérations prévues mais non cruciales, jusqu'à des effets parfois dramatiques³⁶, en passant par des opérations absurdes, des pertes de données ou des dégâts matériels parfois très importants (par exemple dans le cas d'Ariane V outre la destruction de la fusée, la perte de quatre satellites scientifiques de très grandes valeurs).

³⁴ Un logiciel de 500 KLS correspond à 10 volumes de 1000 pages aux normes de l'édition format Pléiade, ce qui fait en moyenne un peu plus de cinq erreurs par volume.

³⁵ On peut distinguer les logiciels critiques "de mission" dont la défaillance est désastreuse sur un plan financier, des logiciels critiques de "sécurité" dont la défaillance peut provoquer des victimes humaines. Ce caractère critique est évidemment relatif : une défaillance du logiciel de gestion d'une entreprise n'est pas critique pour la collectivité, mais peut l'être pour l'entreprise considérée. De même certaines utilisations d'un logiciel peuvent lui conférer un caractère critique inattendu : par exemple, l'emploi (effectivement réalisé) d'un banal tableur par un chirurgien pour traiter des données concernant un patient sur lequel il était en train d'effectuer une opération à cœur ouvert (Lauren Ruth Wiener, 1994, p. 100).

³⁶ En raison d'une séquence d'utilisation non prévue au cours de son développement, une machine à rayons X pilotée par ordinateurs il y environ une dizaine d'années a été la cause de plusieurs accidents mortels dans des hôpitaux (Maarten Boasson, 1998, p. 2).

Dans le cas des progiciels, l'absence de toute garantie de bon fonctionnement de la part des éditeurs de progiciels est significative³⁷. Cette absence totale de garantie n'existe pour aucun autre produit commercialisé. Certes en Europe de telles clauses sont illégales (cf. chapitre III), du moins pour la vente à des particuliers ou à des professionnels non informaticiens, mais il demeure des problèmes d'interprétation notamment autour de la notion de vices cachés, qui ne s'applique que quand il y a vente d'un produit, alors qu'un progiciel n'est jamais vendu : seul est concédé à l'acquéreur un droit d'utilisation.

2 - L'imperfection de procédures qui sont exécutées automatiquement

Comment expliquer cette présence persistante de défauts dans les logiciels ? Pourquoi sont-ils si difficiles à éliminer ? Pourquoi des défauts, en apparence anodins, peuvent avoir des conséquences si dommageables ? La réponse à ces questions nécessite de repartir du rôle d'un logiciel qui pilote un traitement automatique d'informations, et de la contradiction, intrinsèque au logiciel, entre les possibilités de création diversifiée illimitée d'un texte (caractéristique de l'activité d'écriture) et les contraintes résultant du fait que ce texte est un texte actif, un ensemble d'instructions qui seront exécutées automatiquement par des machines (cf. chapitre I). "Un logiciel est une structure composée de symboles logiques arrangés en fonction du modèle que quelqu'un se fait d'un aspect du réel" mais "pour servir à quelque chose, le logiciel doit être intégré à un système physique [et humain] qui affecte une partie du monde" (Lauren Ruth Wiener, 1994, p. 63)³⁸. C'est ce qui explique qu'un système logiciel doit avoir "un niveau de fiabilité très supérieur à celui du système humain, ou de la pratique humaine, que l'informatique remplace ou complète, faute de quoi le système informatique est incompatible avec le rôle social qui lui est dévolu" (Jacques Printz, 1998, p. 18).

³⁷ Par exemple, pour le dernier langage de programmation Delphi proposé par l'éditeur Borland, la "garantie" stipule : "La responsabilité maximum encourue par Borland au titre de la présente garantie est limitée au remplacement du support défectueux et aucune responsabilité n'est encourue pour tout autre dommage sur quelque fondement que ce soit et, notamment, tout dommage indirect et accessoire, même si Borland a été expressément informé de l'éventualité de tels dommages." ; et plus loin en majuscules "AUCUN AUTRE ENGAGEMENT OU GARANTIE N'EST CONSENTI PAR BORLAND ET NOTAMMENT AUCUNE GARANTIE EXPRESSE OU TACITE DE CONFORMITE OU D'ADEQUATION A UN USAGE SPECIFIQUE".

³⁸ Ceci peut permettre de comprendre l'affirmation assez surprenante de Lauren Ruth Wiener : "les logiciels sont bogués parce qu'ils sont construits selon des règles logiques. Ce sont des structures abstraites, complexes et étrangères à notre vécu. Les logiciels sont bogués parce que nous nous en servons pour effectuer une tâche. Ils font donc partie d'un système physique, lui-même vulnérable aux aléas de l'environnement" (1994, p.63).

Dans un système informatique, les composants matériels³⁹ ont atteint une grande fiabilité, permise par l'omniprésence de codes correcteurs d'erreurs (produit des avancées de la théorie du codage), qui permettent d'éliminer les aléas nombreux auxquels sont soumis les circuits⁴⁰. Dans le cas d'équipements critiques, il existe des systèmes à tolérance de panne basés sur la présence de composants redondants, qui permettent de compenser automatiquement la défaillance d'un composant. Cette stratégie est efficace pour des composants matériels qui s'usent selon des processus physiques et que l'on peut tester de manière complète⁴¹. En tant que machine traitant des impulsions électroniques, "la fiabilité [d'un ordinateur] est immense et il est correct de dire que l'ordinateur ne fait quasiment jamais d'erreur" (Philippe Breton, 1990, p. 58).

Mais bien évidemment ces impulsions électroniques, qui représentent des données et des instructions, n'ont aucune signification pour l'ordinateur⁴². Celui-ci exécute mécaniquement les actions prévues par le programme sans en connaître la finalité ou le sens. L'ordinateur ne fait que calculer en suivant machinalement les règles qui lui ont été transmises, toute interprétation lui est impossible, aucune action n'est "aberrante", aucune connaissance n'est implicite. "L'ordinateur est le domaine de la pure raison où tout doit être explicable rationnellement" (Jacques Printz, 1998, p. 317). "L'ordinateur, et par voie de conséquence le raisonnement informatique, ne supporte pas, constitutionnellement, l'ambivalence, l'ambiguïté, c'est à dire la possibilité d'incarner simultanément deux choix

³⁹ Les architectures des éléments matériels sont plus simples et comprennent notamment de nombreuses structures répétitives.

⁴⁰ Déjà en 1967, 90 % des incidents informatiques étaient dus à des défauts de programmation (Lauren Ruth Wiener, 1994, p. 98).

⁴¹ Cette stratégie est par contre peu efficace dans le cas des logiciels. Faire tourner en parallèle plusieurs exemplaires du même programme n'aurait strictement aucun intérêt (pour ce qui concerne les défauts des logiciels), les programmes étant rigoureusement identiques, défauts compris. Faire réaliser plusieurs logiciels par des équipes différentes de programmeurs pour répondre au même problème, et les exécuter simultanément peut sembler judicieux (c'est le cas pour la navette spatiale où le logiciel d'un des cinq ordinateurs, qui effectuent les mêmes traitements, a été réalisé par une autre société). Malheureusement des études ont mis en évidence que si les solutions apportées par des programmeurs différents peuvent être très dissemblables, elles ont tendance à provoquer les mêmes défaillances correspondant à des erreurs identiques concernant les mêmes points délicats (Lauren Ruth Wiener, 1994, p. 166).

⁴² C'est ce qui explique la force et la faiblesse de la vérification automatique d'orthographe comparée à une vérification humaine. Lors d'une vérification automatique, toutes les erreurs dans des mots compliqués dont l'orthographe n'est pas évidente sont toujours corrigées si le mot est présent dans le dictionnaire intégré ; par contre, une erreur de frappe, qui retire toute signification à une phrase, ne sera pas détectée si elle a eu pour conséquence un mot présent dans le dictionnaire.

différents à propos d'une même situation" (Philippe Breton, 1990, p. 82). Tout doit pouvoir être tranché par oui ou par non, toute situation qui peut survenir dans le déroulement du programme doit avoir été explicitement prévue. "Par définition l'ordinateur ne peut réagir qu'aux situations pour lesquelles il a été programmé. Dans le cas le plus favorable, face à une situation imprévue (...), il devra s'arrêter en avertissant l'opérateur humain qui le pilote (...). Dans le cas défavorable, il appliquera une procédure prévue pour une autre situation et n'importe quoi pourra se produire" (Jacques Printz, 1998, p. 35). Il est donc nécessaire *qu'ex-ante* (lors de l'élaboration du logiciel) aient été spécifiées complètement toutes les entrées d'informations possibles (données et instructions) qui pourront se produire *ex-post* (lors de l'utilisation du logiciel), et qu'un traitement adéquat ait été élaboré, pour que les sorties d'information résultant de l'exécution du programme soient celles escomptées par les utilisateurs. Toute la difficulté est de réaliser un "pavage" du domaine d'emploi du logiciel, qui ne laisse pas de "vide" (problème *d'incomplétude* du logiciel, c'est à dire de situations non prévues qui peuvent survenir) et qui ne comporte pas de "chevauchement" (problème *d'ambiguïté ou d'indéterminisme* avec l'existence de deux traitements différents possibles face à une même situation).

Un système informatique n'est pas doté d'un comportement intelligent

Il importe de bien saisir le fossé existant entre le comportement d'un ordinateur, même doté des logiciels les plus perfectionnés et le comportement humain. L'insistance sur les différences de comportement entre l'homme et l'ordinateur est motivée par la place particulière qu'occupe le logiciel à la *frontière* entre le monde humain et le monde machinique. En effet, il est le produit d'une création humaine mais est exécuté automatiquement par une machine. Dans beaucoup de situations, il correspond à une activité antérieurement effectuée par l'homme et qui a été modélisée pour être réalisée automatiquement. Or le fait que *rétrospectivement*, on puisse décrire l'action humaine comme étant réalisée conformément à des règles et en suivant des plans bien établis, peut donner l'illusion que tel est le cas dans la réalité, alors que l'action humaine se caractérise par la manière dont elle traite en souplesse chaque situation nouvelle et imprévue (Harry M. Collins, 1992, p. 66).

Tout d'abord, la puissance du cerveau humain est sans commune mesure avec celle d'un ordinateur ; en se basant sur les estimations (imprécises) de la neurobiologie sur le nombre de neurones et sur le nombre de connexions qui les relie, Jacques Printz calcule que pour

simplement représenter ce réseau il faudrait un rayonnage de disques de 10 gigaoctets (de 2 mètres de hauteur) d'une longueur de 3125 km (1998, p. 41-42). Ensuite l'homme a une vision spatio-temporelle globale de son environnement grâce aux associations autorisées par ces cinq sens, alors que les possibilités de traitement de l'ordinateur ne le rendent apte qu'à une vision parcellaire et linéaire de la réalité⁴³. Par exemple, la reconnaissance d'une photographie parmi 10 000 photographies (ce qui correspond à la capacité estimée de mémorisation du cerveau) prend moins d'une seconde à un être humain, alors que l'exécution d'un programme pour effectuer la même opération par un ordinateur dure plus de dix minutes (idem, p. 20-21). Le principe de fonctionnement du cerveau humain n'a rien de comparable avec celui d'un ordinateur muni de sa programmation. Le cerveau n'est jamais enfermé dans un système de signes. Nous pouvons créer librement les signes dont nous avons besoin, sous la seule contrainte de pouvoir communiquer avec nos semblables. *A la différence des langages informatiques qui sont des systèmes fermés (ou complets), le langage humain fonctionne comme un système ouvert sur le monde extérieur, avec la possibilité de créer de nouveaux concepts.* Nous pouvons effectuer des raisonnements inductifs, ce dont les ordinateurs sont absolument incapables, malgré les discours sur l'existence de prétendues machines à induction dans le cas de certains systèmes experts (Harry M. Collins, 1992, p. 178). Le cerveau entretient en permanence une image spatio-temporelle de la réalité qui nous permet de garder le souvenir du passé (en enregistrant la succession des expériences réussies et des échecs qui jalonnent notre existence), de nous projeter dans le futur, de faire des projets, d'assigner des objectifs à atteindre et de définir des critères de jugement. Notre capacité d'auto-référence (l'image de la réalité que nous construisons fait partie de la réalité) est une source de sens nouveau et de création, qui nous permet en permanence d'ajuster notre comportement à la situation nouvelle qui se crée à chaque instant, par le jeu des essais/erreurs/corrections et en intégrant de manière pro-active de nouvelles informations. Notre univers mental est ouvert et réinterprétable en permanence ce qui explique le non-déterminisme foncier des situations et de l'histoire humaine, à l'opposé du comportement d'un ordinateur qui est et doit rester fondamentalement déterministe (Jacques Printz, 1998, p. 45-46 et 321).

⁴³ Jacques Printz précise pourquoi notre perception immédiate de l'espace rend la programmation si difficile et si contre-intuitive. En effet, les résultats de la programmation sont des textes qui ont pour caractéristique principale d'être unidimensionnels, l'ordinateur ne pouvant faire qu'une lecture séquentielle et linéaire de l'information et des ordres qui lui sont communiqués. Il est donc nécessaire d'opérer "une déconstruction" de la perception globale que nous pouvons avoir du logiciel pour la faire rentrer dans la logique unidimensionnelle de l'ordinateur" (p. 74-75).

Un système informatique n'est pas doté de capacités d'interprétation autonome

Certes il a existé de nombreuses tentatives pour rapprocher dans des situations de travail, le comportement humain du comportement d'un automate : Organisation Scientifique du Travail et taylorisation des tâches, édicition bureaucratique de règles et de procédures devant couvrir tous les cas possibles, mise en place de routines qui "consistent à spécifier complètement *ex-ante* l'ensemble des comportements qui seront requis *ex-post*" (Eric Brousseau, 1997, p. 45). Ces tentatives ont en commun de reposer sur des processus de codification et de formalisation accentuées des différents savoirs humains. Mais ces procédés ont également montré leurs limites : rôle décisif (même si fréquemment sous-estimé) des connaissances tacites non codifiables, notamment en présence d'interactions multiples, importance de l'indéterminisme et de l'ambiguïté comme ressources de l'organisation (Geoffrey Bowker, Susan Leigh Star, 1997, p. 300), "mérites méconnus du flou, et ceux mieux connus du tacite ou de l'implicite" (Olivier Favereau, 1998, p.198), constat que "la simple application d'une règle n'est jamais une application simple" (idem, p. 224).

Contrairement à ce que suppose certaines représentations la rationalité humaine ne se limite pas à un calcul pour trouver la solution optimale, qui dépasse fréquemment les capacités humaines (cf. chapitre VII). L'esprit humain, qui à la différence de l'ordinateur " a plutôt des difficultés à mettre en place un pur raisonnement séquentiel" (Philippe Breton, 1990, p. 82), a par contre des possibilités singulières, plus riches et plus complexes⁴⁴ : les hommes connaissent les finalités des actions qu'ils entreprennent, ils sont impliqués dans l'issue des opérations, ce qui leur permet d'autoréguler leurs comportements (Toran Demarco, Timothy Lister, 1991, p. 123).

En particulier, *ils ont la possibilité d'interpréter les règles* ; or, il existe une "part inéliminable, et parfois capitale, d'interprétation pour exécuter une règle, sans même parler de choisir la règle à exécuter", dans la mesure où "les règles sont toujours incomplètes, en ce qu'elles ne fixent pas exhaustivement leurs conditions d'application" (Olivier Favereau, 1998, p.218). De même, Harry M. Collins souligne que les règles ne peuvent pas contenir les règles de leur propre application, et qu' "il est impossible de prévoir toutes les circonstances d'application d'une règle, (...) cette application [étant] soumise, dans chaque cas particulier,

⁴⁴ A la différence des ordinateurs, "les êtres humains sont très doués pour reconnaître des modèles, pour comprendre la langue naturelle, pour prendre en considération les circonstances extérieures au système, pour faire preuve d'initiative" (Lauren Ruth Wiener, 1994, p. 242).

aux précédents que les hommes sont en droit et en mesure de créer" (1992, p. 130)⁴⁵. Pour avoir une réponse appropriée dans des circonstances imprévues, il est indispensable qu'existe une compréhension des règles et une "sensibilité socialisée au contexte grâce à laquelle nous savons quand et avec quelle fréquence il est possible de transgresser les règles" (idem, p. 273)⁴⁶. Olivier Favereau énonce trois caractérisations de ce que signifie "*interpréter*" (en tenant compte de l'existence de capacités d'apprentissage) et qu'il oppose à "*calculer*". Ces caractérisations nous semblent particulièrement fécondes pour comprendre la différence entre le comportement d'une machine programmée par un logiciel, qui reste "prisonnière" d'un modèle fermé constitutif (indépendamment de l'éventuelle complexité de la modélisation réalisée), et le comportement humain. Par sa capacité à interpréter, l'homme peut "élaborer des modèles décidant simultanément de ce qui est général (traitable par le modèle) et de ce qui est particulier (extérieur au modèle)" (idem, p. 199). Il peut "rentrer à l'intérieur d'un système de règles (au sens le plus large) pour lui prêter sens et pertinence, tout en restant à l'extérieur de celui-ci, afin de l'appréhender dans un jugement critique" (idem, p. 205-206). Enfin, nous pouvons "nous mettre à la place d'autrui, non pour anticiper, de l'extérieur ses réactions que pour nous donner toutes les chances de communiquer nos intentions, sans équivoque, et ainsi comprendre de l'intérieur, le sens profond de ses réactions" (idem, p. 220).

Un système informatique ne peut réagir correctement à l'imprévu

Une autre différence importante est la réaction à l'imprévu. L'ordinateur ne pouvant réagir qu'à ce pourquoi il a été programmé, l'imprévisible ne peut lui être que fatal (Jacques Printz, 1998, p. 102), alors que l'homme dispose de capacités d'adaptation à l'imprévu, en ayant notamment la possibilité de réagir en "temps réel" au vu des conséquences des décisions qu'il prend, et des actions qu'il entreprend. C'est ce qui explique les conséquences différentes d'une défaillance psychocognitive et d'une défaillance d'un programme informatique. "Notre

⁴⁵ Harry M. Collins précise qu'il est toujours nécessaire d'établir de nouvelles interprétations, qui ne peuvent pas s'inspirer de l'expérience passée. Il montre que même "dans le cas de sports bien établis, avec des règles organiques hautement codifiées, appliquées à un très haut niveau avec l'organisation rigoureuse convenant aux compétitions de niveau mondial, les règles [sont] incapables par elles-mêmes de faire face à des circonstances imprévues" (p. 130), ce qu'il illustre par le cas, lors de la série de tirs au but pour départager le Brésil et la France en quarts de finale de la Coupe du Monde de football de 1986, d'un tir qui rebondit sur un poteau puis sur le gardien avant de rentrer dans le but (p. 118).

⁴⁶ Harry M. Collins cite le contre-exemple d'un conducteur d'autobus suspendu parce qu'il avait suivi trop strictement une règle (ne pas s'écarter de son itinéraire sans autorisation), alors qu'une personne avait une crise cardiaque dans le bus qu'il conduisait. Il considère qu'il s'agit d'un cas manifeste de "socialisation déficiente en matière de priorités morales" (1992, p. 115-116).

système psychocognitif est une fantastique "machine" tolérante à un nombre considérable de pannes" (idem). En particulier, en cas d'erreur, nous tentons de restaurer l'environnement initial, ce qui est impossible pour l'ordinateur face à une situation imprévue, et qui peut troubler profondément l'utilisateur, compte tenu de son modèle psychocognitif (idem, p. 127).

Enfin, l'ordinateur est évidemment dépourvu de deux qualités importantes de l'être humain, le bon sens et l'intuition. "Le bon sens (...) est constitué du stock de situations dont nous disposons à l'instant t pour juger du bien-fondé de ce que nous sommes en train d'effectuer à l'instant $t+\Delta t$. Le bon sens fonctionne comme une procédure d'exception, un démon de Maxwell, qui observe en permanence nos actions et qui est en mesure d'en déclencher l'arrêt si celles-ci, eu égard au passé, sont en mesure de nuire à notre intégrité présente et future. Sans le bon sens nous aurions probablement disparu de la planète depuis bien longtemps" (Jacques Printz, 1998, p. 34). Quant à l'intuition c'est une "qualité spontanée dont on ne contrôle ni la présence, ni l'absence, mais qui semble jouer un rôle permanent – parfois diffus, parfois aigu – dans toute activité humaine" et "aucun raisonnement produit par le cerveau humain ne peut d'ailleurs éviter de mettre en œuvre l'intuition" (Philippe Breton, 1990, p. 61)⁴⁷.

L'utilisation d'un système informatique requiert fréquemment des "dialogues" entre les utilisateurs et l'ordinateur, qui sont pilotés par les logiciels. Aux incertitudes sur le comportement des utilisateurs s'ajoutent les incertitudes sur l'environnement sur lequel l'ordinateur capte des informations et sur lequel éventuellement il agit, d'autant plus que ces environnements se modifient par rapport au moment où a été conçu le logiciel. Le problème principal est celui de la prédictibilité : tous les possibles doivent être prédéterminés, précodés par le programmeur, alors qu'il "reste presque toujours une marge de virtualités indéterminées, un espace libre pour la création de possibles que nul n'avait prévus" (Pierre Lévy, 1992, p. 27)⁴⁸. Comme toute interprétation des règles est impossible, les règles doivent être plus complètes, plus cohérentes, plus détaillées, pour éviter que leur application aveugle ne conduise à des absurdités : relance de clients pour zéro franc, poursuite d'un contribuable

⁴⁷ Philippe Breton ajoute que même dans le cas des informaticiens qui affirment n'utiliser qu'une démarche logique, l'observation de leurs comportements montrent que, en réalité, les informaticiens recourent fréquemment à une démarche intuitive (1990, p. 67).

⁴⁸ Comme le note Harry M. Collins, si "on peut avoir une science sociale causale tournée vers le passé, on ne peut faire une science sociale prédictive et déterministe" (1992, p. 174).

pour trois centimes (Jean-louis Peaucelle, 1997, p. 27). Il faut en particulier se méfier des situations qui n'apparaissent pas ambiguës à un être humain, parce qu'elles renvoient pour lui à des connaissances élémentaires⁴⁹, quasi naturelles (comme le fait que l'année qui suit l'année 1999 n'est pas l'année 1900 !), mais qui le seront pour l'ordinateur, si ces informations élémentaires n'ont pas été explicitement introduites. De plus, le logiciel de par sa nature immatérielle apparaissant facilement modifiable, la tentation peut être forte de modifier plus fréquemment les règles, ou de les complexifier pour traiter des situations particulières (qui, traitées "manuellement" n'auraient pas nécessité l'édiction de nouvelles règles). Cependant, la croyance, qu'en rajoutant règle après règle à des programmes pour répondre aux exceptions et aux cas particuliers découverts on finira par reproduire le comportement humain et qu'il ne sera plus nécessaire d'effectuer des modifications ultérieures, est vaine : "réussir à intégrer quelque chose rétrospectivement dans une règle ne veut pas dire que la règle pourra rendre compte d'exemples futurs imprévus" (Harry M. Collins, 1992, p. 67)⁵⁰.

Les conséquences sur les difficultés à développer des programmes "corrects"

Une des principales difficultés dans l'écriture d'un programme correct, consiste à réaliser un traitement adéquat pour tous les cas qui pourront se présenter, pour toutes les combinaisons de toutes les entrées d'informations possibles. Il faut, en général, plus d'effort pour traiter et vérifier la validité du traitement des quelques cas qui se présentent exceptionnellement que pour la grande majorité des situations habituelles. Cela nécessite de la part de l'informaticien de "pousser l'utilisateur dans ses derniers retranchements pour obtenir de lui toutes les réactions à prévoir face à toutes les situations envisageables", alors que l'utilisateur a tendance, "fort pragmatiquement, à laisser dans l'oubli ces cas de figures irréalistes voire invraisemblables, relevant à l'évidence d'un traitement pour période de crise, sur lequel il serait toujours temps de se pencher le cas échéant" (Jean-Marie Desaintquentin, Bernard Sauter, 1991, p. 11). De plus la pleine collaboration des utilisateurs n'est pas nécessairement acquise : méfiance face à une "obsession inquisitoriale, révélatrice de cette

⁴⁹ "Un problème, bien connu des anthropologues et des chercheurs en sciences humaines qui ont un cadre de pensée interprétatif, est l'invisibilité de ce qui nous est trop familier" (Harry M. Collins, 1992, p. 202-203).

⁵⁰ Ce même auteur souligne que "l'incapacité à faire des progrès est souvent imputée, non pas à la nature fondamentale du savoir, mais à des défauts que l'on n'a pas vus dans l'établissement des règles. Les programmeurs et les cognitivistes tentent de faire en sorte que les règles reflètent la réalité de plus en plus précisément, mais la précision ne désigne pas la qualité la plus essentielle d'une bonne performance. C'est une erreur que d'essayer de contrefaire le savoir-faire sur la base d'un surcroît de précision" (Harry M. Collins, 1992, p. 282-283).

volonté de pouvoir qui, selon [l'utilisateur], anime les informaticiens" (idem, p. 12), crainte d'une critique des procédures suivies dans la réalité et parfois éloignées des normes théoriques en vigueur, inquiétude par rapport à une dévalorisation éventuelle de ses compétences résultant de leur transfert dans un logiciel⁵¹.

Ce problème concerne également l'introduction des données par les utilisateurs. Il faut distinguer l'intégrité formelle des données (assurée par un contrôle automatique et systématique des données entrées par rapport à des fourchettes de valeurs autorisées⁵²) et l'intégrité réelle (la validité du contenu même de l'information), qui ne peut être garantie que par l'utilisateur. Or, la saisie des données n'est jamais purement technique mais toujours de nature sociotechnique (Philip E. Agre, 1997, p. 258), la détention d'informations étant un des fondements du pouvoir des acteurs. En particulier, la pertinence des données entrées peut être problématique, lorsque les personnes qui les saisissent ne sont pas celles qui les utiliseront après traitement (Geoffrey Bowker, Susan Leigh Star, 1997, p. 292). Peuvent s'enclencher des cercles vicieux de données erronées entrées, de sorties d'informations résultantes non fiables, et de désintérêt pour saisir des données correctes.

En conséquence, "les systèmes informatiques, dans la mesure où il cohabite avec des humains, ou fonctionnent dans des environnements hostiles, doivent être dotés de mécanismes compensateurs et/ou réparateurs permettant de répondre aux fluctuations de l'environnement externe et aux aléas internes, tous imprévisibles par définition" (Jacques Printz, 1998, p. 18)⁵³.

Cette analyse peut permettre de comprendre *pourquoi les logiciels comprennent des erreurs et pourquoi celles-ci sont si difficiles à éliminer* : elles peuvent se manifester

⁵¹ Nous reviendrons plus en détail dans la section suivante, consacrée aux difficultés d'adaptation du logiciel aux besoins, sur l'importante question de la collaboration, pas nécessairement spontanée des utilisateurs.

⁵² Il faut se montrer très prudent dans la restriction des données acceptées en entrée d'un programme, surtout quand la validité des données ne dépend pas de conventions humaines (une note ou un prix) mais de l'observation de phénomènes physiques. Par exemple, dans les années soixante-dix et quatre-vingt, le logiciel qui traitait les relevés des taux d'ozone recueillis par les satellites de la Nasa, les rejetait en les considérant comme des mesures erronées parce qu'ils étaient trop bas, avant qu'une équipe de scientifiques britanniques en 1986 ne signale la chute du niveau d'ozone (Lauren Ruth Wiener, 1994, p.36).

⁵³ Dans le cadre d'une analyse plus générale des systèmes artificiels, Herbert Simon souligne que "nous aurons souvent à nous satisfaire d'approximation pour accorder la conception aux objectifs. Dans de tels cas, les propriétés du système interne "transparaîtront" à travers sa frontière - autrement dit, le comportement du système répondra seulement de façon partielle aux sollicitations de l'environnement, et il s'ajustera partiellement aussi aux limitations du système interne" (1974, p. 28).

seulement après des années d'utilisation des logiciels⁵⁴, il est souvent difficile d'isoler au sein de systèmes de plus en plus interconnectés le logiciel défectueux⁵⁵ surtout lorsque les défaillances se produisent de façon fugace et intermittente et qu'il est difficile de reproduire le contexte d'apparition de l'erreur⁵⁶.

3 - La détection et la correction des erreurs

De nombreuses étapes dont l'objectif est d'éliminer les défauts jalonnent les processus de développement des logiciels. Capers Jones distingue jusqu'à treize niveaux d'élimination des défauts : revue de spécifications, revue de conception générale, revues de conception détaillée, revues d'assurance-qualité, validations et vérifications faites par des indépendants, preuves de correction, tests sur table, tests unitaires, tests fonctionnels, tests de composants, tests du système, tests sur le terrain, tests de réception par le client (Capers Jones, 1989, p. 207-208)⁵⁷.

Les différentes opérations d'élimination des défauts ont un rendement très inégal, mais qui est difficile à évaluer, dans la mesure où on ne connaît les défauts non trouvés qu'a posteriori. En plus des efforts pour trouver les défauts, il faut prendre en compte le coût des réparations. Ce coût peut être très élevé quand la vérification du système logiciel à l'aide de tests met en évidence des erreurs de spécification et de conception. Or les principales sources d'erreurs des logiciels, surtout pour les logiciels de taille importante, sont beaucoup plus fréquemment des problèmes de spécification et de conception, que des problèmes de codage. C'est pourquoi dans les années soixante-dix et quatre-vingt se sont développées des

⁵⁴ Il a été constaté pour certains logiciels que le nombre d'erreurs découvertes, qui décroît logiquement durant la phase d'utilisation de ce logiciel, pouvait se remettre à croître en conséquence d'une plus grande expertise des utilisateurs qui l'exploitaient pour des opérations plus complexes dans des environnements de plus en plus diversifiés.

⁵⁵ Les défaillances fréquentes lors d'une connexion sur un serveur distant peuvent provenir d'une erreur dans un des nombreux logiciels concernés (systèmes d'exploitation de l'ordinateur client et du serveur, logiciel de navigation sur le réseau, système de gestion des données sur le serveur, logiciels assurant la transmission des données...) sans qu'il soit aisé de déterminer lequel.

⁵⁶ Une enquête sur un système d'exploitation très répandu a démontré que 30 % des bogues signalées présentaient un risque d'apparition d'une fois tous les cinq mille ans de fonctionnement du système (Lauren Ruth Wiener, 1994, p. 129).

⁵⁷ En intégrant la documentation et les autres fournitures qui accompagnent un logiciel (matériel d'accompagnement et de formation), Capers Jones liste plus de quarante variétés de méthodes d'élimination des défauts qui peuvent être utilisées (Capers Jones, 1989, p. 212-223).

techniques de validation d'un logiciel qui "s'intéressent à la spécification externe du logiciel que l'on considère alors comme une boîte noire" (Jacques Printz, 1998, p. 351). Ces méthodes jugées extrêmement rentables sont des techniques d'inspection et de revue des documents issus du cycle de développement.

Mais même en appliquant ces méthodes, la phase la plus importante reste la phase des tests, qui peut représenter jusqu'à la moitié du temps total de développement d'un logiciel (Frederick P. Brooks, 1996, p. 16)⁵⁸. Paradoxalement, "le passé de l'informatique n'a pas mis en avant le caractère essentiel de cette tâche" (Serge Bouchy, 1994, p. 177), pour laquelle il existe peu d'enseignements théoriques. Cette activité longue, fastidieuse et coûteuse reste "la bête noire" de l'informatique (idem), voire la "honte de la communauté informatique, tant au niveau industriel, qu'au niveau universitaire" (Jacques Printz, 1998, p. 249)⁵⁹.

Le test est ce qui valide *in fine* la bonne correspondance entre le modèle informatique que l'ordinateur exécute et la réalité. Il consiste à effectuer une exécution sur machine de scénarios d'essais, conçus comme des protocoles expérimentaux parfaitement reproductibles, ce qui exige un certain déterminisme du système observé, l'élimination progressive des erreurs pouvant être assimilée à la falsification expérimentale d'une théorie selon Popper (Jacques Printz, 1998, p. 104).

On distingue les tests "boîte blanche" où on vérifie pas à pas les différents "chemins" possibles dans le programme, des tests "boîte noire" où on ignore volontairement le détail du programme pour ne s'intéresser qu'à la correspondance entre état initial et état final. Cette dernière stratégie peut être extrêmement coûteuse le nombre de correspondances possibles connaissant rapidement une "combinatoire explosive" et le coût des tests augmentant probablement de façon exponentielle (Jacques Printz, 1998, p. 258).

C'est une des raisons qui a justifié de recourir à une conception modulaire des logiciels, avec la constitution de modules les plus indépendants possibles qui pourront être testés

⁵⁸ Frederick P. Brooks précise qu'en général cette importance n'est pas celle qui a été prévue initialement, mais celle que l'on constate *a posteriori*. Par exemple, les coûts des tests pour la navette spatiale américaine ont constitué plus de 60 % du coût total des logiciels (Jacques Printz, 1998, p. 110).

⁵⁹ Les réponses à une enquête de 1985 indiquait que moins de 14 % des entreprises faisaient état de mesures de la qualité du logiciel ou des éliminations de défauts (Capers Jones, 1989, p. 40).

séparément (tests unitaires). Il faut éviter le plus possible le "couplage" entre modules⁶⁰, car dans ce cas la fiabilité de la somme fonctionnelle de deux modules est égale au produit des fiabilités (Jacques Printz, 1998, p. 287)⁶¹. Néanmoins il faudra ensuite réaliser des tests d'intégration avec les autres composants du système, dans toutes les combinaisons possibles, dont le nombre croît très vite. Ce test prend énormément de temps, car "des bogues subtils peuvent résulter d'interactions inattendues entre des composants déjà débogués" (Frederick P. Brooks, 1996, p. 5)⁶². "Face à de telles combinatoires, seuls la puissance du cerveau et le savoir-faire humain sont à même de donner des résultats sémantiquement pertinents. Les tests sont donc une activité à haute valeur sémantique ajoutée où, *très rapidement, le cerveau humain est irremplaçable*" (Jacques Printz, 1998, p. 259)⁶³.

Aux difficultés pour détecter les erreurs, s'ajoutent les problèmes occasionnés par la réparation des erreurs constatées. En effet, "en corrigeant un bogue on a une chance substantielle (20 à 50 %) d'en introduire un autre" (Frederick P. Brooks, 1996, p. 103), une des explications étant que fréquemment ceux qui corrigent les logiciels sont moins compétents que ceux qui ont écrit le code initial. De plus, "toute correction tend à détruire la structure du système, à augmenter son entropie et son désordre" (idem), ce qui est d'autant plus vrai que le logiciel a été conçu longtemps auparavant, par des personnes qui ne sont plus nécessairement dans l'entreprise et que sa documentation n'est pas forcément de qualité. Dans ces situations, les personnes chargées de la maintenance résolvent le problème par des "colmatages, c'est-à-dire des corrections locales inadéquates et des rafistolages de fortune" (Lauren Ruth Wiener, 1994, p. 138). C'est notamment ce qui explique que les derniers défauts découverts sont toujours les plus coûteux à éliminer (Capers Jones, 1989, p. 25). Dans la mesure où les effets de la correction d'une erreur peuvent être lointains et indirects, il est donc

⁶⁰ Des modules sont faiblement couplés quand les échanges entre modules sont peu nombreux.

⁶¹ Quand la fiabilité globale est le produit des fiabilités élémentaires, elle tend inéluctablement vers zéro dès que le nombre d'éléments devient grand (Jacques Printz, 1998, p. 132).

⁶² C'est, par exemple, une des difficultés pour corriger le bogue de l'an 2000 : l'existence de deux logiciels, qui testés séparément passent le cap de l'an 2000, ne garantit pas que leur utilisation simultanée ne posera pas de problème.

⁶³ Philippe Breton constate qu'il est souvent nécessaire de recourir à l'intuition pour retrouver une erreur dans un logiciel (1990, p. 61).

nécessaire de refaire de nombreux tests (tests de régression)⁶⁴. De ce fait, le coût du passage de n tests croît en n^2 , chaque nouveau test accompagné d'une correction nécessitant une vérification de conformité des tests déjà acquis, soit au total $n(n+1) / 2$ passages (Jacques Printz, 1998, p. 259).

En conclusion, il faut prendre conscience qu' "un test n'est pas destiné à montrer qu'un programme fonctionne correctement, mais à trouver des erreurs" (Jean-Marc Geib, 1989, p. 99). "Certaines études ont permis d'estimer que le nombre d'erreurs de programmation présentes au début des tests est couramment de l'ordre de 2 pour 10 lignes de code. Pour un programme de 100 000 lignes, c'est donc 20 000 erreurs, de gravité heureusement très variable, qui devront être décelées et corrigées" (Gérard Dréan, 1996, p.216). En général, on estime qu'après les tests, seulement 85 % des défauts existants ont été éliminés.

B - DES ERREURS DE MOINS EN MOINS ACCEPTEES...

La présence de défauts dans les logiciels est difficile à accepter car elle résulte d'erreurs purement humaines (1), alors même que leurs conséquences sont de plus en plus sensibles avec l'augmentation de l'importance des logiciels (2).

1 - Des erreurs humaines difficiles à accepter...

Un logiciel est un artefact. Tout objet artificiel créé par l'homme peut bien évidemment comporter des défauts. Mais la singularité des logiciels est que ce sont des artefacts complètement immatériels, qu'ils sont constitués exclusivement des produits de l'activité humaine seule⁶⁵. De ce fait, l'origine des défauts est toujours une erreur humaine, cette erreur ayant pu se produire en phase de conception/programmation, en phase

⁶⁴ Un logiciel de routage d'appels comportant plusieurs millions de lignes de codes et dont les tests avaient duré treize semaines, subit une très légère modification (elle ne concernait que trois lignes). Les programmeurs, jugeant qu'ils connaissaient parfaitement les conséquences de cette modification, estimèrent qu'il était inutile de refaire les tests. Cette modification minimale comprenait une erreur (un "6" à la place d'un "D"), qui entraîna une série de coupures pour les abonnés au téléphone de plusieurs grandes villes américaines fin juin et début juillet 1991 (Lauren Ruth Wiener, 1994, p. 27 et 67).

⁶⁵ "Les bugs sont une panne d'un genre inédit dans l'histoire des techniques. Ils ne sont pas le fruit de conditions extérieures, comme l'oxydation, qui grippe un mécanisme. Ils ne sont pas non plus le résultat d'une pièce qui se brise. C'est une sorte d'oubli de la part des concepteurs de logiciels, incapables d'envisager tous les cas de figure de l'utilisation de leurs produits. Le bug est le résultat d'une limite des pouvoirs de l'homme, de sa capacité d'abstraction et de sa faculté d'anticiper des scénarios d'utilisation et des combinaisons de technologies sans cesse plus complexes" (Thibault Honnet, interview à Libération-Multimédia, 31-12-1999).

d'installation/maintenance ou en phase d'exploitation. "Il est donc impossible d'attribuer à une quelconque fatalité ce qui est totalement imputable aux concepteurs et aux programmeurs du système" (Jacques Printz, 1998, p. 57). Par contre, dans le cas d'un artefact matériel, un défaut peut résulter d'une "imperfection de la nature", avec la présence de matériaux imparfaits dans l'artefact. C'est ce qui explique qu'un manque de fiabilité au niveau du matériel soit mieux accepté qu'au niveau du logiciel, alors que la grande majorité des problèmes informatiques sont des problèmes logiciels, et que les rares défaillances des composants matériels sont actuellement le plus souvent la conséquence d'erreurs dans les logiciels (microcode) qu'ils intègrent⁶⁶.

Il semble que la société tolère moins bien les imperfections humaines que les imperfections naturelles. "Les limites de la science" et les "limitations d'ordre social" sont des évidences récentes et mal intégrées dans le corps social (Jacques Printz, 1998, p. 13). Accepter l'existence d'erreurs humaines – tout en cherchant à en limiter les conséquences – semble difficile et constitue une rupture radicale avec un "système éducatif qui nie radicalement l'erreur et qui surtout la culpabilise, rendant ainsi sa prise en compte de façon sereine extrêmement problématique" (idem, p. 19-20). Il est pourtant inévitable que dans une activité aussi exigeante que la programmation se manifestent des "défaillances du processus psychocognitif"⁶⁷ dont la conséquence sera une erreur humaine. Un problème supplémentaire provient du fait que quand une défaillance logicielle se manifeste, si celle-ci est inévitablement le produit d'une erreur humaine, il peut être très difficile (et dans certains cas impossible) de l'imputer à un individu ou à un collectif d'individus précis : la responsabilité est distribuée, au sens fort, entre l'ensemble des agents impliqués (Laurent Thévenot, 1997, p. 236).

Cette difficulté à reconnaître que ce sont des erreurs humaines qui sont à l'origine des défauts des logiciels, se retrouve dans le terme choisi pour les désigner : les *bugs* (que l'on a tenté de franciser sans grand succès en bogues). C'est une mathématicienne, Grace Murray Hopper, qui travaillait sur le Mark 2 de l'Université de Harvard le 9 septembre 1947, qui décida d'appeler "*bug* tout ce qui empêche un programme de marcher", après que l'on eut

⁶⁶ Un exemple récent est le cas de la première génération de microprocesseurs "Pentium" commercialisés, dont le programme intégré pour effectuer les divisions sur des nombres à virgule flottante, comportait une erreur.

⁶⁷ Jacques Printz a établi une nomenclature impressionnante de la variété des défaillances psychocognitives qui peuvent se manifester dans l'activité de conception de logiciels (p. 101).

constaté qu'une défaillance sur un de ces premiers ordinateurs s'expliquait par la présence d'un papillon de nuit dans les circuits de l'appareil⁶⁸. Il est significatif, qu'alors que la présence intempestive de ces créatures naturelles ne peut plus expliquer des défaillances informatiques depuis plusieurs dizaines d'années, ce soit une expression qui fait disparaître toute référence humaine au problème, que l'on continue à utiliser.

2 - ...avec l'augmentation de l'importance des logiciels

Ces erreurs sont de plus en plus difficiles à accepter dans la mesure où leurs conséquences concernent directement et indirectement de plus en plus de monde. L'environnement socioéconomique est de plus en plus dépendant du bon fonctionnement de nombreux systèmes informatisés (téléphone, contrôle aérien, TGV, billetteries, futur système informatisé pour la santé). Quand ces systèmes rendent des services en ligne au grand public, les pannes sont jugées insupportables (Jacques Printz, 1998, p. 260). Les erreurs en informatique ont toujours eu "la particularité d'arriver à tout bloquer rapidement" (Philippe Breton, 1990, p. 37), mais les risques sont plus graves avec l'interconnexion grandissante de l'ensemble des systèmes informatiques. En outre, la complexité croissante de ces systèmes augmente considérablement les coûts des réparations : Jacques Printz estime que le coût de réparation d'un système suite à une défaillance peut être très supérieur à 10 hommes/année d'effort pour une seule réparation (1998, p. 52). Les dépenses engagées par les entreprises et les administrations pour que leurs systèmes passent l'an 2000 (en raison du codage des années sur deux chiffres), sont délicates à évaluer dans la mesure où une part importante de ces dépenses consiste à moderniser des systèmes, une modernisation qui aurait été effectuée ultérieurement en l'absence du bogue de l'an 2000. Au niveau mondial, les estimations se situent entre 300 et 600 milliards de dollars selon le Gartner Group (250 milliards de dollars selon IDC) ; les 330 plus grandes entreprises américaines ont dépensé 22,8 milliards de dollars selon une enquête du Wall Street Journal (3/1/2000), le record étant atteint par le groupe bancaire Citigroup qui, à lui seul, a dépensé 950 millions de dollars.

En même temps, le fait que le marché informatique, longtemps dominé par l'offre, soit de plus en plus déterminé par la demande, fait que les utilisateurs sont de moins en moins disposés à supporter les conséquences des défauts logiciels. Par exemple, le groupe Toshiba a

⁶⁸ Elle ajouta que c'était "le premier cas réel de bug à avoir été trouvé" dans la mesure où cette expression était déjà utilisée pour désigner des pannes du télégraphe électrique depuis son origine.

provisionné un milliard de dollar pour éviter un procès visant un dysfonctionnement du logiciel pilotant les lecteurs de disquettes de ses ordinateurs portables susceptible d'entraîner des pertes de données, ce problème concernant plus de 5 millions d'ordinateurs vendus aux Etats-Unis depuis 1985. D'autres procédures judiciaires visant des grands noms de l'informatique (Packard Bell, filiale de NEC, Hewlett-Packard, Compaq...) sont en cours (18h.com le quotidien de l'Expansion, 2-11-99).

Cependant, le problème des défauts logiciels ne va pas s'estomper car ils sont de plus en plus difficiles à éliminer.

C - ...MAIS DE PLUS EN PLUS DIFFICILES A ELIMINER

1 - Des logiciels de plus en plus complexes dans des environnements de plus en plus diversifiés

Certes des progrès substantiels ont été effectués par rapport à l'époque des débuts de l'informatique, où Jean-Marie Desaintquentin et Bernard Sauter parlaient de l'état à peine avouable des logiciels de base quand ils arrivaient chez le client et où la mise au point était achevée sur le site (1991, p. 10). Deux facteurs importants de l'existence de nombreux défauts durant cette période ont disparu. Il s'agit d'une part du manque de ressources des premières générations d'ordinateurs qui avait créé une culture de la programmation fondée sur la recherche du codage des données et des algorithmes le plus compact possible (Jacques Printz, 1998, p. 136), et d'autre part, de l'utilisation de langages de programmation de bas niveau. Mais en même temps, les progrès réalisés dans ces domaines ont entraîné l'apparition de nouveaux problèmes : les langages de haut niveau permettent de masquer la complexité de l'architecture matérielle sous-jacente et leur utilisation permet d'effectuer moins d'erreur lors de la conception des logiciels, mais lorsqu'il se produit une défaillance difficilement explicable du système, il est nécessaire de bien comprendre ce que fait le système jusqu'à des niveaux de détails architecturaux masqués par le langage utilisé. De plus la rapidité des changements dans le secteur informatique empêche fréquemment d'atteindre la phase de maturité, où "le taux de défaillance par unité de temps est stable, ce qui veut dire qu'il y a autoadaptation entre le système, les usagers et les utilisateurs" (Jacques Printz, 1998, p. 346). Surtout, la formidable augmentation des ressources matérielles mises à disposition a entraîné

la création de logiciels d'une taille⁶⁹ et d'une complexité de plus en plus importante (cf. chapitre III).

Or, "le nombre d'erreurs croît de façon non linéaire avec la taille des systèmes [et] leurs fréquences d'apparition augmentent avec la puissance des installations" (Jacques Printz, 1998, p. 11)⁷⁰. De plus, un aspect critique pour la fiabilité des logiciels est qu'ils constituent un des composants d'un système composé d'autres logiciels, d'unités centrales et de périphérique. Avec l'explosion de la microinformatique et l'existence de périphériques et de logiciels de plus en plus divers, il existe une "infinie variété de configurations de matériels et de logiciels, assemblés de façon personnalisée en systèmes adaptés à chaque utilisation particulière" (Gérard Dréan, 1996 A, p. 311). Des incertitudes supplémentaires viennent de la mise en réseau progressive de l'ensemble de ces équipements et de leur ouverture sur le monde extérieur, ce qui les rend particulièrement sensible aux "agressions" extérieures⁷¹. De plus en plus de systèmes, fonctionnant en temps quasi réel, tendent à s'interconnecter (systèmes boursiers et bancaires, systèmes de transports et de réservations de plus en plus divers). Les interactions se multiplient entre de plus en plus d'acteurs (hommes, organisations, ordinateurs, réseaux, capteurs, actionneurs) avec des procédures et des règles à respecter toujours plus nombreuses. S'appliquent les conclusions des travaux d'Ashby : plus on augmente le nombre de connexions dans un système, plus celui-ci tend à devenir instable (cité par Pascal Petit, 1998, p. 405)⁷².

⁶⁹ Un logiciel d'un million de lignes de code source est devenu courant.

⁷⁰ "Quand vous augmentez de 50 % la taille d'un programme, il faut 100 ou 150 % de temps en plus pour le déboguer" (Jon Shirley de Microsoft, cité par Frédéric Dromby, 1999, p. 640).

⁷¹ Ces "agressions" peuvent dans certains cas être intentionnelles. Il s'agit alors de "virus" informatiques, métaphore partiellement adéquate pour désigner la propagation de petits programmes malveillants : avec l'interopérabilité des systèmes on peut parler de disparition des "barrières immunitaires" existantes. Toutefois, un virus informatique est toujours une création humaine. Selon le cabinet de recherche Computer Economics, pour l'année 1999 c'est plus de 12 milliards de dollars qui, à l'échelle mondiale, sont allés à la prévention contre les virus, à la remise en état des systèmes et à la compensation des pertes de données occasionnées par ces attaques (Les Chroniques de Cybérie, 18-1-2000).

⁷² Dans un livre empreint d'un certain catastrophisme ("La bombe informatique", 1998), Paul Virilio fait une analogie entre l'interactivité qui est à l'information ce que la radioactivité est à l'énergie, en précisant que les risques de la première sont plus importants ("l'émergence redoutable de "l'Accident des accidents", p. 148) vu son caractère global, alors que la radioactivité n'est que locale.

Enfin, l'amélioration de la "convivialité" des applications développées s'accompagne également d'un fort accroissement de complexité⁷³. Dans le même temps, le passage des architectures centralisées, où les traitements s'effectuent sur un mainframe avec de simples moniteurs transactionnels pour les usagers, à des architectures clients-serveurs où les traitements sont répartis sur l'ensemble des machines connectées et où les interactions sont multidirectionnelles, crée de nouvelles difficultés pour localiser les défauts responsables d'une défaillance. L'ensemble de ces évolutions contribue à la multiplication des défauts "diffus" à la différence des défauts localisés physiquement dans un programme bien précis, qui en général déclenchent une panne franche facilement détectable et reproductible. A l'inverse les défauts diffus engendrent fréquemment des pannes intermittentes et fugaces, résultant d'une combinaison d'actions très diverses, et sont donc très difficiles à reproduire, ce qui rend la correction de ces erreurs particulièrement laborieuse et coûteuse (Jacques Printz, 1998, p. 58).

On peut ajouter qu'une cause de complexité supplémentaire, qui fut la source de nombreux problèmes, provient paradoxalement des mesures de sécurité qui conduisent pour des missions critiques à ajouter au logiciel de multiples fonctions de sécurité, ou à faire exécuter les mêmes traitements par plusieurs ordinateurs différents (jusqu'à cinq pour la navette spatiale) pour le cas où un ordinateur tomberait en panne (Lauren Ruth Wiener, 1994, p. 56 et 170).

"Jamais dans son histoire, l'homme n'a fait face à un tel niveau de complexité. Outre les erreurs de conception, les erreurs de manipulation sont inévitables. L'environnement du système est plus ou moins bien contrôlé, voire même incontrôlable. Tout le problème est donc, non pas d'éviter les erreurs - c'est désormais définitivement impossible - mais de s'organiser pour vivre avec, et de limiter les dégâts qu'elles peuvent occasionner" (Jacques Printz, 1998, p. 19).

2 - L'impossibilité des logiciels sans défaut et ses conséquences

"Les informaticiens, à la différence des non initiés savent que le logiciel sans défaut est impossible" (OCDE, 1991 A, p. 9). "Toute personne bien informée des questions de génie logiciel et de formes plus classiques d'ingénierie pourra vous confirmer que l'état de l'art en

⁷³ Jacques Printz estime que la demande irrépressible en facilité d'emploi et en convivialité s'accompagne d'une croissance au minimum exponentielle de la complexité du logiciel (1998, p. 200).

génie logiciel est loin d'atteindre l'état de développement d'autres secteurs d'ingénierie. Lorsqu'un produit d'ingénierie classique a été réalisé, testé et mis sur le marché, on peut raisonnablement penser qu'il a été correctement conçu et qu'il fonctionnera de manière fiable. Il est normal, en revanche, qu'un produit logiciel contienne des bogues importantes et ne fonctionne pas de manière satisfaisante. (...) Les professionnels du logiciel (...) savent que les programmeurs les plus compétents ne peuvent éviter ce genre d'incidents" (David L. Parnas, préface à Lauren Ruth Wiener, 1994, p. XV). Cette impossibilité, tout du moins pour des programmes d'une certaine importance, peut être montrée sur un plan théorique (a), mais elle résulte surtout de contraintes économiques (b).

a - Une impossibilité théorique

Cette impossibilité s'explique par la double nature d'un logiciel comme étant un texte numérique actif (cf. chapitre I). Un logiciel est un texte numérique et sa création s'effectue sur un support d'une souplesse inégalée : "peu de supports de création se montrent si flexibles, si faciles à effacer et à retravailler, si aptes à accueillir de grandes structures conceptuelles" (Frederick P. Brooks, 1996, p. 6). Il permet au pouvoir de l'imagination de s'exercer pleinement, "le programmeur, comme le poète, [maniant] des abstractions voisines de la pensée pure" (idem). Ceci se retrouve dans l'extraordinaire diversité des logiciels qui sont écrits, tant au niveau de la variété des problèmes qu'ils tentent de résoudre, que de la diversité des solutions proposées pour un même problème⁷⁴. Mais un logiciel est également un "texte qui agit" dans la mesure où il se compose d'un ensemble d'instructions qui seront exécutées automatiquement par une machine. "La construction du programmeur, au contraire des vers du poète, est réelle : elle agit, produisant des résultats distincts de la création elle-même" (idem). On peut distinguer dans toute activité créatrice trois étapes : l'idée, l'implémentation et l'interaction. Quand une création s'effectue sur un support matériel, la puissance de l'imagination est bridée par les contraintes de l'implémentation. Ce n'est pas le cas du logiciel où c'est seulement au moment de son interaction que se manifesteront les imperfections inévitables d'une activité humaine⁷⁵. Le logiciel, au stade de sa conception est "libre des

⁷⁴ "Pour résoudre un même problème, vous ne trouverez jamais deux programmeurs proposer la même solution. Un programme est un miroir qui reflète l'esprit qui l'a produit, et les esprits humains sont très dissemblables" (Lauren Ruth Wiener, 1994, p. 120).

⁷⁵ Frederick P. Brooks ajoute deux arguments : le fait que le programmeur dépend tant pour la conception que pour l'exécution de son programme, des programmes d'autrui (qui également ne sont pas parfaits), et le fait

contraintes du sens commun et des lois de la physique" (Lauren Ruth Wiener, 1994, p. 64), mais dans son utilisation, comme tout artefact, il n'est pas "hors de la nature" et n'a "nulle dispense pour ignorer ou pour violer les lois de la nature" (Herbert Simon, 1974, p. 17). Par contre, on peut considérer que "la programmation, et plus encore l'analyse, reste malgré tout un art, un art rigoureux et logique, certes, mais un art quand même, qui résiste en tant que tel à tout algorithme" (Philippe Breton, 1987, p. 216), et il est malheureusement impossible de démontrer que le produit de cet art se comportera "correctement"⁷⁶.

Les limites des vérifications formelles des programmes

Certes, se sont développées des techniques de vérification formelle des programmes, parfois également appelées "preuves de correction". Il existe deux grandes familles de méthodes formelles qui ont en commun un substrat mathématique fort. Les méthodes constructives ou logiques sont inspirées des mathématiques axiomatisées qui garantissent qu'en appliquant des axiomes et des règles logiques de raisonnement, on engendre nécessairement des théorèmes vrais. Appliquées au logiciel, ces méthodes consistent, à partir des spécifications initiales du logiciel, à démontrer rigoureusement tous les raisonnements, qui par des procédés de raffinements successifs, conduisent au programme final. La plus représentative de ces méthodes est la méthode B, qui a succédé à la méthode Z, appliquée notamment à certains modules logiciels de sûreté dans les transports. Les autres méthodes sont les méthodes par vérification de modèles, qui reposent sur le fait que les programmes peuvent se modéliser à partir d'objets mathématiques aux propriétés bien définies : les automates. La vérification du programme consiste à explorer à l'aide d'algorithmes l'espace des états de l'automate résultant de toutes les interactions possibles. Leur mise en œuvre nécessite impérativement d'utiliser des langages de programmation spécifiques (Lustre, Esterel⁷⁷...) qui ne permettent de construire que des automates, ce qui n'est pas le cas des langages de programmation classiques comme ADA, C, ou C++ (Jacques Printz, 1998, p. 111-112).

qu'il "est amusant d'élaborer de savants concepts, alors qu'extirper de petits bogues de son code n'est qu'un labeur ingrat" (1996, p. 7).

⁷⁶ Philippe Breton rappelle opportunément que Turing a "inventé" l'ordinateur, non pour calculer mais pour prouver les limites du calcul ; cela lui a servi à démontrer, entre autres, qu'il n'existait pas d'algorithme capable de démontrer qu'un autre algorithme pouvait résoudre un problème donné (1987, p. 216-217).

⁷⁷ Ces langages sont utilisés pour programmer des systèmes temps réels synchrones, fréquents en avionique ou dans les protocoles de communication.

Les limites de ces méthodes sont connues. La première de ces limites tient aux limites de la logique mathématique. Kurt Gödel a démontré en 1931, à partir de travaux sur les systèmes formels de l'arithmétique, que tout système complexe contient des énoncés indécidables, c'est à dire indémontrables si l'on demeure dans le formalisme du système considéré. Un système formel est nécessairement soit incomplet (une théorie est complète si toutes les propositions logiquement valides que l'on peut former avec les termes de cette théorie sont ou vraies ou fausses), soit incohérent (une théorie est cohérente ou consistante lorsque l'on ne peut pas démontrer un théorème et son contraire). Il est certes nécessaire de s'assurer que l'emploi qui est fait d'un modèle est "à la fois consistant et complet, mais ce sont malheureusement des propriétés indécidables par des algorithmes" et c'est "le cerveau humain, et lui seul, qui *in fine* pourra prononcer la validité de la construction logique, car lui seul est maître de la correspondance avec le réel" (Jacques Printz, 1998, p. 169).

En effet, ces méthodes ne peuvent au mieux que vérifier la conformité du programme aux spécifications initiales. Or celles-ci sont difficiles à établir⁷⁸ et peuvent omettre certains phénomènes extérieurs au système perçus comme des évidences. La spécification initiale est un modèle de la réalité⁷⁹ dont la validité ne peut être établie qu'expérimentalement. "Le processus d'abstraction, par la simplification qu'il impose, permet de passer du monde réel à des abstractions mathématiques idéalisées, mais ce processus n'est pas lui-même mathématique, et sa validité n'est pas du domaine de la preuve" (Jacques Printz, 1998, p. 113). Ces méthodes concernent le domaine de la validité (ou adéquation ou correction) du logiciel, c'est à dire l'aptitude du système à assurer exactement ses fonctions définies dans le cahier des charges, qu'il importe de différencier de la fiabilité (ou robustesse ou tolérance) qui désigne l'aptitude du système à fonctionner même dans des conditions anormales (Jean-Marc Geib, 1989, p. 20). Jacques Printz ajoute : "les figures du monde réel, comme les programmes

⁷⁸ "Il est presque toujours difficile pour l'expert humain de décrire le savoir dans des termes précis, complets et suffisamment cohérents pour pouvoir être utilisés dans un programme d'ordinateur. Cette difficulté vient de la nature inhérente du savoir qui constitue l'expertise humaine : il est souvent subconscient et peut-être approximatif, incomplet et incohérent". Cette citation de Buchanan et al. (1983) effectuée par Harry M. Collins, (1992, p. 125) à propos du problème de l'élucidation du savoir pour construire des systèmes experts, nous semble pouvoir être généralisée à l'élaboration des spécifications d'un logiciel.

⁷⁹ "Les modèles informatiques ne peuvent être que limités. Comme l'information numérisée, leur résolution est finie. On ne peut représenter qu'un certain nombre de propriétés, prendre en compte qu'un nombre limité de paramètres et ne calculer les résultats qu'à un nombre précis de décimales. Tous les modèles sont plus ou moins subtils, mais ils sont toujours moins subtils que l'infinie complexité de la réalité" (Lauren Ruth Wiener, 1994, p. 197).

concrets ne sont pas des objets mathématiques, mais ce qui permet de les dessiner et de les écrire peut l'être ; cela donnera à l'objet réel et concret un certain nombre de caractéristiques logiques minimales éventuellement démontrables. Quant à la réalité proprement dite, autant vouloir "démontrer" qu'un moteur n'explosera pas, ou qu'un pont ne s'effondrera pas, à la seule vue des équations qui gouvernent le phénomène physique caractéristique de l'artefact" (1998, p. 68).

Une autre limite de ces méthodes est l'importance du travail qu'elles requièrent pour être appliquées, ce qui les rend très coûteuses et fait qu'elles ne peuvent être utilisées que pour des composants logiciels de taille limitée⁸⁰. Leur efficacité est controversée : Frederick P. Brooks, même s'il reconnaît que la vérification de programmes ne veut pas dire qu'on obtient des programmes sans erreur⁸¹, estime que ces méthodes peuvent réduire (mais non supprimer) les tests à effectuer (1996, p. 169). Par contre, selon Capers Jones, aucune étude n'a montré que la fiabilité opérationnelle des programmes ayant subi des preuves de correction, était supérieure aux autres programmes (1989, p. 214 et 215).

L'influence de l'environnement

Une deuxième raison de l'impossibilité théorique des logiciels sans défauts résulte du fait qu'une défaillance d'un programme peut provenir soit de l'algorithme du programme uniquement, soit de la conjonction de l'algorithme et de son comportement dans un environnement donné. Or si un algorithme est généralement déterministe (sauf s'il est faux !), ce n'est pas le cas du comportement d'un programme qui est fondamentalement non déterministe. On oublie fréquemment qu'un ordinateur n'est pas rigoureusement déterministe,

⁸⁰ Le coût est encore plus élevé si l'on tente de combiner la vérification formelle du logiciel avec des méthodes pour élaborer des spécifications proches des utilisateurs. David Gianazza et alii (1997) ont ainsi développé pour un système de contrôle de trafic aérien, un premier système à partir des spécifications des utilisateurs, puis ils ont procédé à une rétro-conception pour aboutir à une spécification formelle en Z permettant une validation du modèle. Celle-ci a permis de détecter des incomplétudes, des ambiguïtés et quelques contradictions, invalidant les spécifications initiales, ce qui nécessitait de reprendre tout le processus de développement à son point de départ. Les auteurs en concluent que l'application conjointe des deux méthodes semi-formelle et formelle, si elle est particulièrement intéressante, ne peut être utilisée que pour certains éléments et pas pour un système dans son ensemble.

⁸¹ Frederick P. Brooks ajoute aux arguments déjà mentionnés le fait que même les démonstrations mathématiques peuvent être fausses.

est une machine à états discontinus⁸² - alors que la plupart des phénomènes physiques sont continus - et qu'il ne respecte pas toujours les règles logiques de l'arithmétique élémentaire, la logique ignorant le temps et la durée ce qui n'est pas le cas des ordinateurs⁸³. Le non-déterminisme des ordinateurs est aggravé par deux caractéristiques importantes des systèmes informatiques : la recherche d'un partage maximal des opérations et des ressources ; l'asynchronisme et la non-instantanéité des opérations effectuées par la machine⁸⁴. L'importance de ces facteurs augmente avec l'existence de programmes de plus en plus dynamiques et interactifs⁸⁵. En fonction de la durée de la session, du niveau d'interactivité (nombre d'utilisateurs connectés), de la capacité du programme à créer dynamiquement les entités nécessaires à son exécution et de la vitesse relative des différentes unités qui interagissent au sein d'une même machine, il se produit une dégradation inévitable de l'environnement et l'apparition de "fantômes" (Jacques Printz, 1998, p. 117). L'existence de ces "fantômes", qui ne peuvent être détruits que par l'arrêt physique de la machine et la réinitialisation de toutes ses structures de travail et du système d'exploitation, n'a par définition pas été prévue lors de l'écriture d'un programme. Elle rend la détermination des causes de certaines défaillances, et donc la correction des erreurs, particulièrement difficiles, puisque l'on ne peut reproduire exactement le contexte d'apparition de la défaillance. Une solution consiste à rajouter des programmes observateurs, appelés "démons" dont le rôle consiste à noter tous les états intermédiaires du processus de traitement de façon à faciliter le diagnostic en cas de défaillance. Cette solution n'est toutefois pas parfaite car les programmes "démons"

⁸² Un ordinateur "est une machine à états discrets, ne pouvant traiter que des informations de type digital, à l'exclusion par exemple des informations continues, qui sont pourtant dominantes dans la nature et forment sans doute l'une des bases du raisonnement humain" (Philippe Breton, 1987, p. 163).

⁸³ Même si les performances des ordinateurs sont extrêmement spectaculaires, il faut intégrer le fait qu'avec le temps les logiciels peuvent être utilisés dans des conditions qui dépassent les marges de sécurité prévues dans les spécifications initiales. Par exemple, le logiciel installé sur le système de défense antimissile Patriot générerait une erreur d'un millionième de seconde toutes les dix secondes pour tracer la fenêtre de poursuite des missiles. Cette erreur restait sans conséquence tant que le système était utilisé pendant moins d'une journée sans interruption ce qui correspondait au cahier des charges de l'armée américaine. Par contre pendant la guerre du golfe ce système fut utilisé pendant cinq jours sans interruption. De ce fait, l'erreur n'était plus négligeable et fut responsable de la non-interception d'un missile irakien le 25 février 1991 qui tua vingt-huit soldats américains et en blessa quatre-vingt-dix-huit autres (Lauren Ruth Wiener, 1994, p. 31-32).

⁸⁴ Si le système était synchrone, il devrait fonctionner au rythme de son organe le plus lent. Or il existe de très grandes différences de vitesse d'exécution entre les différents organes d'un ordinateur.

⁸⁵ Ces facteurs sont particulièrement critiques dans le cas de traitement en parallèle où le programme exécute des instructions sur plusieurs processeurs simultanément.

consomment eux-mêmes des ressources, en proportion de l'importance des observations que l'on veut effectuer, et perturbent de ce fait l'observation de l'état réel du système⁸⁶.

En conséquence, "dans l'état actuel des technologies de programmation, le seul moyen de vérifier qu'un programme est correct est de l'exécuter" (Gérard Dréan, 1996 A, p. 216)⁸⁷. C'est ce que réalisent les tests. Mais l'exécution des tests, même très nombreux, ne permet pas de certifier qu'un programme ne contient pas d'erreur. D'une part, il faudrait énumérer tous les cas possibles de défaillance des modules et de leurs combinaisons, puis procéder aux vérifications correspondantes à l'aide de test. Les combinaisons induites étant en n puissance n , les coûts et les délais de réalisation deviennent rapidement impraticables (Jacques Printz, 1998, p. 110). D'autre part, les programmes de tests et les jeux d'essais peuvent eux-mêmes contenir des erreurs⁸⁸ : Capers Jones cite une étude informelle menée à IBM en 1975 qui montrait que la densité des erreurs des données de test d'un programme était supérieure à celle du programme lui-même (1989, p. 216).

On ne peut donc jamais certifier qu'un programme ne contient plus d'erreur, mais plus modestement dire "que la probabilité de détecter une erreur existante croît avec la durée des tests, et que la probabilité qu'il existe des erreurs résiduelles est d'autant plus faible que le système a fonctionné sans incidents pendant plus longtemps dans une plus grande variété de configurations" (Gérard Dréan, 1996 A, p. 218).

Cette impossibilité d'une fiabilité totale des logiciels dans toutes les circonstances a des implications sur la conception des logiciels, qui va à rebours de l'utopie technologique dominante, pour laquelle "tout changement, toute modification d'une situation, toute résolution de problème, passe désormais par une solution technologique, notamment en

⁸⁶ La présence du "démon" fausse l'observation ; ce qui est en fait observé c'est le système plus son "démon". On peut établir une analogie avec l'observation des phénomènes microphysiques, où le comportement externe observé est la somme des comportements du phénomène proprement dit, et de celui de l'observateur.

⁸⁷ Cela rejoint le constat que l'analyse théorique ne peut se passer d'expériences (Herbert Simon, 1974, p. 36-37).

⁸⁸ De façon plus générale une défaillance, même en phase d'utilisation n'est pas nécessairement perçue comme telle, si le programme fournit des données en sortie erronées mais plausibles. Lauren Ruth Wiener cite le cas d'un logiciel qui fournissait des prévisions incorrectes sur les marges brutes de ventes et donc sur les bénéfices. Ce n'est qu'à la fin de l'année que l'entreprise se rendit compte que son bénéfice estimé à 500 000 dollars était en fait une perte de 4 millions de dollars, ce qui contraignit la société à déposer son bilan (1994, p.36).

termes d'information et de communication" (Philippe Breton⁸⁹). Par exemple, dans le domaine des systèmes experts, il est illusoire et il peut être dangereux de vouloir remplacer des experts par des machines ; il est préférable de limiter les ambitions des systèmes experts au rôle d'aide des experts humains (Harry M. Collins, 1992, p. 236). De façon plus générale, il faut, quand c'est possible, prévoir des possibilités d'intervention humaine qui contournent les automatismes, pour traiter les cas particuliers qui n'auront pas tous été prévus par les concepteurs⁹⁰. Pour les systèmes critiques, il faut envisager systématiquement les conséquences d'une défaillance toujours possible de ces systèmes, ce qui peut conduire à limiter les rôles de tels systèmes⁹¹. Notamment, il importe de se poser la question de la comparaison entre la situation résultant d'une défaillance (toujours possible) d'un système et la situation caractérisée par l'absence de système. L'impossibilité pour pouvoir tester dans des conditions réalistes les logiciels de l'initiative de défense stratégique (IDS), ce qui aurait nécessité de procéder à des explosions nucléaires à l'air libre, a motivé l'opposition à ce projet du professeur David L. Parnas, membre du comité de réflexion sur les aspects informatiques de l'IDS, pour qui il est impossible qu'un système logiciel de grande taille fonctionne correctement dès son premier usage⁹².

⁸⁹ in "L'utopie de la guerre technologique", *Le Monde*, 30 avril 1999.

⁹⁰ Ces possibilités reposent sur la conception que "la machine va plus vite que l'homme pour traiter de la même façon les mêmes informations, tandis que l'homme, dont le raisonnement est plus que de la "simple" logique, recèle bien d'autres possibilités que celles de la machine" (Philippe Breton, 1990, p. 92). Elles supposent l'entretien des compétences humaines pour effectuer des opérations, qui faute de pratique risquent de s'atrophier (Abbe Moshowitz, 1997, p. 35-36). Dans de nombreux cas, elles nécessitent l'existence d'une "traçabilité" des traitements automatiques qui ont échoué, pour comprendre les raisons de ces échecs. Dans le domaine de l'aviation, Lauren Ruth Wiener cite le cas de la protection d'enveloppe installée sur la plupart des systèmes numériques de pilotage et qui garantit que les commandes de l'avion ne franchiront pas des limites mettant en cause l'intégrité de la structure de l'avion. Dans certaines situations quasi désespérées, il peut être préférable que le pilote puisse dépasser ces limites (1994, p. 117 et 167).

⁹¹ Par exemple, dans le cas du traitement par radiations où un défaut logiciel fut responsable de plusieurs décès, le simple maintien des systèmes de sécurité mécaniques qui équipaient les générations antérieures d'appareils non automatisés, aurait suffi à éviter les conséquences dramatiques de la défaillance du logiciel. De même, un logiciel destiné à économiser le carburant en ralentissant au maximum les moteurs sur les Boeing 767 fut responsable d'un vol plané de 14 minutes lors de la descente vers un aéroport. La température extérieure était exceptionnellement basse de sorte que le ralentissement des moteurs entraîna la formation de glace qui provoqua l'arrêt des moteurs. Le paramètre de la température extérieure, critique dans ce cas précis, n'avait pas été pris en compte par les concepteurs du logiciel (Lauren Ruth Wiener, 1994, p. 71).

⁹² Il est amusant de noter qu'un autre expert, Solomon J. Buchsbaum, directeur aux laboratoires ATT Bell, consulté par le congrès des Etats-Unis sur les logiciels destinés au projet d'IDS, estima que l'on pouvait concevoir un tel système avec une fiabilité garantie, la preuve en ayant été apportée par système logiciel du réseau public américain de télécommunications... quatre années avant que ce système soit responsable d'une impossibilité de communiquer pour la moitié de la clientèle d'ATT (Lauren Ruth Wiener, 1994, p. 141).

b - Les contraintes économiques

Cependant, affirmer qu'une fiabilité parfaite est impossible ne signifie nullement que tous les logiciels aient des niveaux de fiabilité équivalents. Ce qui va jouer un rôle déterminant ce sont les arbitrages économiques différents qui vont être effectués entre la fiabilité et les autres caractéristiques du logiciel et de son processus de production. Nous avons signalé à de nombreuses reprises le temps et le coût très élevé, pour détecter et corriger les défauts, facteurs qui augmentent fortement lorsque le nombre de défauts restants diminue.

Les impératifs d'une production marchande de logiciels font prédominer "la contrainte qu'un système doit être développé avec un coût convenu à l'avance et dans un délai donné" (Maarten Boasson, 1998, p. 4). Dans la pratique, une entreprise fixe un calendrier pour des raisons de marché ou de concurrence et laisse les erreurs qui sont découvertes trop tard en les signalant et en les corrigeant dans une prochaine version (Gérard Dréan, 1996 A, p. 218). De même, Pierre Bonnaure estime qu'une des raisons importantes de la présence de bogues dans les logiciels est leur commercialisation prématurée (1996 B, p. 66). Ce phénomène est particulièrement présent pour les développeurs de logiciels destinés au "grand public", qui "travaillent le plus souvent sous une telle pression que les produits qu'ils livrent sont criblés d'imperfections" (Lauren Ruth Wiener, 1994, p. 98). Dans la mesure où les tests constituent le plus souvent la dernière étape du processus de développement, il est fréquent que la phase de test soit raccourcie en raison des dépassements de calendrier des étapes précédentes. Outre les problèmes de délais, qui, en raison de l'importance des rendements croissants d'adoption dans ce secteur, peuvent être décisifs, il faut prendre en compte les coûts de développement : Barry W. Boehm indique que la productivité s'effondre si l'on recherche une fiabilité extrême comme dans le cas du logiciel de commandes de la navette spatiale développé par IBM (Frederick P. Brooks, 1996, p. 188)⁹³. Katia Messika, dirigeante d'une entreprise spécialisée dans le test des logiciels (*International Testing*), précise qu'elle teste les produits sur cinquante machines, qui correspondent à 80 % du parc informatique⁹⁴ et ajoute que, si "tous

⁹³ Ce logiciel a coûté environ mille dollars la ligne, alors que la moyenne se situe entre vingt-cinq et cent dollars. Malgré cela, en se basant sur le nombre de bogues découvertes dans des versions antérieures, le producteur lui-même, IBM, estime qu'il contient encore une cinquantaine de bogues (Lauren Ruth Wiener, 1994, p.155).

⁹⁴ Ceci implique qu'un utilisateur sur cinq risque de se retrouver confronté à des bogues dus à un conflit entre le programme et le matériel. Il faut de plus ajouter que ces tests sont faits sur des machines "vierges" et ne peuvent donc intégrer les bogues résultant d'interactions avec d'autres programmes installés par l'utilisateur.

les bugs que nous signalons sont corrigeables, tous ne sont pas corrigés" (*Libération-Multimédia*, 31 décembre 1999)⁹⁵. En effet, en intégrant le fait que ces défauts ne sont pas visibles au moment de l'achat du logiciel et que les préjudices éventuels ne sont généralement pas à la charge de l'entreprise productrice, la présence de défauts peut ne pas avoir que des conséquences négatives pour certains producteurs placés en situation de quasi-monopole : revenus provenant du service après vente⁹⁶, incitation à acheter les nouvelles versions qui corrigeront notamment certains des bogues constatés⁹⁷.

Selon Capers Jones, "la seule façon de se rapprocher d'un rendement de 100 % dans l'élimination des défauts est de tourner en production sur une longue période avec l'ensemble des clients, bien que ce soit là une méthode coûteuse, qui cause souvent beaucoup de soucis à l'équipe de développement et une certaine exaspération chez les utilisateurs de programmes" (1989, p. 217)⁹⁸. De même, Lauren Ruth Wiener, estime que "rien n'est plus efficace pour mettre au jour des bogues qu'une utilisation massive pendant des années dans des conditions réelles et impitoyables" et que "l'usage réel par des utilisateurs réels permet de découvrir plus de bogues que n'importe quel programme de tests" (1994, p. 132)⁹⁹. Ce type de pratique, qui suppose une coopération en partie désintéressée des utilisateurs, est difficilement praticable dans un cadre marchand avant la commercialisation du produit. Si l'on souhaite de plus que certains utilisateurs puissent non seulement signaler les défauts rencontrés mais également proposer des solutions à ces défauts, cela suppose de diffuser le code source du logiciel ce qui rend difficile la vente du logiciel (cf. chapitre VII).

⁹⁵ Selon Mark Minasi (2000), "90 % des bugs signalés par l'utilisateur final étaient déjà connus de l'éditeur avant la mise en vente".

⁹⁶ Ces revenus peuvent être très importants, notamment quand l'entreprise qui commercialise le logiciel ne diffuse pas son code source (cas le plus fréquent) ce qui la met en situation de monopole pour le service après vente du logiciel.

⁹⁷ Précisons également que les nouvelles versions si elles corrigent des anciens bogues identifiés peuvent en introduire de nouveaux, notamment quand l'introduction de nouvelles fonctionnalités aux effets secondaires mal identifiés, est ajouté rapidement au produit sans reconcevoir et retester l'ensemble du logiciel.

⁹⁸ Une pratique proche pour les logiciels consiste à proposer à certains utilisateurs une version dite *bêta* (déjà testée par les développeurs mais non encore commercialisée), en espérant que ces utilisateurs transmettront les bogues qu'ils constateront.

⁹⁹ Pour les systèmes techniques en général, Jacques Girin note que "les réseaux de relations qui aboutissent à des systèmes techniques n'ayant pas d'utilisateurs réels et contemporains – mais seulement des utilisateurs potentiels et futurs – ont toutes les chances d'être défailants" (1994, p. 28).

En fait, il faut lier la question de la fiabilité du logiciel aux autres caractéristiques du logiciel, avec lesquelles les relations peuvent être contradictoires : la rapidité d'exécution du logiciel (une plus grande fiabilité consommant des ressources qui peuvent être importantes¹⁰⁰), la portabilité du logiciel (avec les contraintes supplémentaires provenant de sa possibilité d'exécution sur des machines différentes), la "convivialité" (source d'une complexité additionnelle). Les différents producteurs de logiciels effectuent des compromis qui peuvent être très variables entre ces différentes caractéristiques, et qui ne sont pas toujours en faveur de la fiabilité. Pour certains utilisateurs et certains types d'utilisation la fiabilité n'est pas nécessairement la caractéristique principale, d'autant plus qu'en cas de dysfonctionnement (pas toujours perceptible¹⁰¹) il peut être difficile de savoir quel est le produit incriminé.

Section III - Des logiciels inadaptés aux besoins des utilisateurs ?

Il s'agit ici d'examiner, au-delà de leur fiabilité, dans quelle mesure les logiciels sont conformes aux attentes des utilisateurs, indépendamment du fait que ces attentes aient été formellement exprimées. C'est le problème plus général de la qualité, au sens classique du terme, qui est posé. En effet, l'AFNOR qui définissait la qualité d'un produit comme "son aptitude à satisfaire les besoins des utilisateurs" (définition AFNOR NF X 50-109), a modifié cette définition en juillet 1982, pour préciser qu'il s'agissait "des besoins exprimés ou implicites" (AFNOR NF X 50-120)¹⁰². Cette précision est importante pour les logiciels où "la qualité des applications informatisées s'apprécie autant, voire même plus, par la capacité permanente de compréhension et d'adaptation des informaticiens à la problématique des utilisateurs, que par la conformité des spécifications ou des produits résultant simplement de la bonne application d'une démarche professionnelle maîtrisée" (Serge Bouchy, 1994, p. 171). En ce domaine, la principale difficulté consiste à "passer de l'expression imprécise d'un besoin

¹⁰⁰ C'est pour cette raison que les logiciels de commande de processus (comme les systèmes de contrôle aérien, ou de pilotage d'une centrale nucléaire), qui sont des logiciels critiques de sécurité, sont particulièrement difficiles à développer dans la mesure où ils exigent simultanément un haut niveau de fiabilité et une grande vitesse d'exécution pour réagir en temps quasi réel.

¹⁰¹ Un des reproches adressé au système d'exploitation de Microsoft est qu'il laisse des morceaux de mémoire non récupérés après fermeture des applications, ce qui à terme dégrade les performances. Ceci n'est pas directement perceptible pour un utilisateur non expert.

¹⁰² De même la norme internationale ISO 8042 définit la qualité comme "l'ensemble des caractéristiques d'une entité qui lui confère l'aptitude à satisfaire des besoins exprimés et implicites".

utilisateur, à la mise en place d'un produit sans faille, activable par des utilisateurs divers dans des conditions multiples et qui correspond aux attentes opérationnelles des usagers" (idem, p. 29).

De ce point de vue, l'insatisfaction des utilisateurs à l'égard des logiciels apparaît comme importante, même si nous verrons qu'il est nécessaire de relativiser le constat effectué, la qualité étant plus une construction sociale qu'une réalité objective (A). En première analyse cette qualité perçue comme insuffisante peut s'expliquer par une relation initialement très déséquilibrée entre les informaticiens et les utilisateurs (B). L'amélioration du rapport de forces en faveur des utilisateurs a contribué, entre autres, à des progrès significatifs dans la qualité des logiciels, mais la diversité croissante des logiciels et de leurs utilisateurs, et les exigences de la concurrence – qui peut ne porter qu'accessoirement sur la qualité des produits – font qu'il existe une insatisfaction permanente des utilisateurs (C). Celle-ci peut également s'expliquer par les spécificités du logiciel (D).

A - LE CONSTAT D'UNE INADAPTATION APPAREMMENT PERMANENTE AUX BESOINS DES UTILISATEURS

1 - Un constat...

Selon Serge Bouchy, "le bon logiciel n'est pas seulement celui qui, *in fine*, répond exactement à l'énoncé d'origine, mais c'est surtout celui qui satisfait l'utilisateur final, au moment de son emploi, dans des conditions de coûts et de délais qui correspondent aux ressources de l'entreprise concernée" (1994 p. 175). Selon cette conception, le bilan n'apparaît pas comme étant particulièrement positif : "les concepteurs de logiciels sont souvent accusés d'être en retard sur les projets et de ne pas apporter les fonctions promises. (...) [il existe] une impression persistante que l'utilité et la facilité d'utilisation (...) sont souvent surestimés par les fournisseurs, les utilisateurs devant s'accomoder de systèmes inappropriés" (OCDE, 1993, p. 114).

Même s'il faut considérer avec prudence des estimations statistiques toujours partielles, la proportion des applications qui ne sont jamais utilisées ou qui nécessitent d'importantes modifications est impressionnante. Une étude célèbre des services de la comptabilité gouvernementale américaine effectuée en 1979 sur neuf grands projets de l'administration américaine en tirait un bilan désastreux :

Tableau XXXXI
Bilan des logiciels de neuf grands projets de l'administration américaine
(en milliers de dollars et en pourcentage de l'ensemble)

	Montant	Pourcentage
Logiciels utilisés tel que livrés	118	2%
Logiciels utilisés après modifications	196	3%
Logiciels utilisés mais refondus ou abandonnés plus tard	1300	19%
Logiciels payés mais non livrés	1950	29%
Logiciels livrés mais jamais utilisés avec succès	3200	47%
Total	6764	100%

Source : Etude des services de la comptabilité gouvernementale américaine, citée par Patrick Jaulent (1992, p. 16)

Frédéric Georges Roux à partir d'une enquête de satisfaction auprès des entreprises françaises estime que 53 % des applications sont "inadéquates et à refaire" (1991, p. 39). Concernant les systèmes d'information, Marie-Christine Monnoyer-Longe note que les échecs sont de 20 à 50 % selon les sources, et estime que l'insatisfaction des utilisateurs dépasse 30 % (1997, p. 118). Les fonctionnalités disponibles ne sont que très partiellement utilisées (Pierre Bonnaure, 1996 B, p. 67) ; dans le cas des progiciels c'est seulement 20 % en moyenne de ces fonctionnalités dont se servent les utilisateurs (Eurostaf, 1996 C, p. 79). L'étude des services des grandes organisations françaises montre que dans de nombreux cas, l'informatique est moins vécue comme une aide que comme un "casse-tête supplémentaire" (Serge Bouchy, 1994, p. 182). Cette qualité insuffisante ne concerne pas que le fonctionnement du logiciel, mais également sa documentation : celle-ci est jugée difficile à comprendre dans 75 % des cas et contient des erreurs dans la moitié des cas ; elle manque le plus souvent d'"exemples qui marchent vraiment" (Capers Jones, 1989, p. 217-218).

De façon plus générale, Philippe Breton (1990) estime que le développement de l'informatique a toujours été fondé sur la promesse de services à venir et non sur les services réellement rendus.

Une autre façon d'apprécier la qualité plus faible des logiciels que des autres produits, est la mesure de la maturité des processus de développement de logiciels. Celle-ci peut être

effectuée à partir d'un modèle, *Software Process Maturity Model* (SMM)¹⁰³ élaboré en 1991 par le *Software Engineering Institute* (SEI) de l'Université de Carnegie Mellon. Ce modèle se base sur les mêmes principes que la série des normes ISO 9000, c'est à dire que la qualité du produit est appréhendée à partir de la qualité du processus de fabrication. Il analyse une organisation selon cinq axes (organisation de projet, ingénierie du produit, gestion quantitative, amélioration continue et technologie utilisée) et dix-huit domaines d'activité principaux (planification de projet, gestion des spécifications, assurance qualité...). Le modèle distingue cinq niveaux de maturité (initial, répétitif, défini, géré, optimisé) ; seul un nombre infime d'organisations atteignent le niveau 3, les niveaux 4 et 5 étant "des territoires quasiment inexplorés en ce qui concerne le logiciel" alors que "les caractéristiques de ces niveaux ont été définies par analogie avec ce qui s'observe pour d'autres types d'industrie" (Luc Rubiello, 1997, p. 232).

2 - ...qu'il convient de relativiser

Toutefois il importe de relativiser ce constat négatif et le fait qu'il serait de la responsabilité exclusive des producteurs de logiciels.

La responsabilité des utilisateurs

Tout d'abord, à côté des reproches des clients-utilisateurs aux prestataires, il est nécessaire de prendre en compte les reproches des prestataires à leurs clients, notamment dans la formulation des besoins. A partir de l'observation de projets ayant dû subir des modifications, Capers Jones remarque que "les utilisateurs eux-mêmes ne savaient pas toujours très bien ce qu'ils voulaient réellement que fasse le système" (1989, p. 273). Frederick P. Brooks estime "qu'il est en fait impossible aux clients, même assistés d'ingénieurs logiciels, de spécifier correctement, précisément et complètement les détails d'un système logiciel moderne avant d'avoir construit et essayé au moins une version du produit qu'ils spécifient" (1996, p. 172). Le constat de cet auteur est tranché : "il faut bien le dire, les clients ne savent pas ce qu'ils veulent. Ils ne savent pas en général quelles questions il faut poser, et ils n'ont souvent jamais pensé au problème avec le niveau de détail qu'il faut pour écrire les spécifications. Même les réponses simples ("faites marcher le nouveau logiciel comme notre ancien système de traitement manuel") sont en fait trop simples. Les clients ne

¹⁰³ Parfois également appelé *Capacity Maturity Model for Software* (CMM).

veulent jamais dupliquer exactement l'existant. En outre, les systèmes logiciels complexes sont des choses qui agissent, qui bougent, qui travaillent. La dynamique de ces actions est difficile à imaginer" (idem).

Le problème, analysé par Faïz Gallouj (1994, p. 194 et suivantes) dans le cas plus général des activités de services, est celui de l'écart existant entre le besoin "réel" de l'utilisateur à partir duquel il va évaluer la qualité de la prestation fournie, et le besoin "exprimé" par l'utilisateur et "perçu" ou "ressenti" par le prestataire qui va déterminer la prestation réalisée. Comme le note Serge Bouchy, dans le cas des logiciels il est "toujours difficile de faire formaliser clairement par les utilisateurs finals (...) l'expression de la qualité explicite" (1994, p. 179). Pour étudier les différents agencements organisationnels, Jacques Girin souligne l'asymétrie fondamentale dans une relation de mandat, entre le mandant qui "dit" et le mandataire qui "fait" (1994, p. 18). Jacques Girin distingue quatre situations possibles selon que le "dire du mandant" (ou "mandement") est clair ou confus, et selon que l'activité du mandataire est simple (facile à décrire) ou complexe (difficile à décrire). On peut considérer que dans l'activité de productions de logiciels, on se trouve le plus souvent dans la situation la plus problématique, d'un mandat confus et d'une activité complexe, où "le mandataire ne maîtrise pas tous les éléments conditionnant la réussite de la prestation, laquelle dépend aussi de la coopération et des capacités du mandant" (idem, p. 20).

La complexité de l'activité fait qu'il est souvent difficile pour le mandant de comprendre que "de petites variations de la description du système peuvent engendrer de grandes variations de l'effort de développement et/ou des ressources nécessaires à l'exécution du système" (Jacques Printz, 1998, p. 190). Les utilisateurs ne soupçonnent pas notamment les conséquences que peuvent avoir des exigences accrues en termes de "convivialité" des applications (Jean-Marie Desaintquentin, Bernard Sauter, 1991, p. 16)¹⁰⁴. Maarten Boasson note que "les maîtres d'ouvrage, faute d'une bonne compréhension, peuvent avoir des attentes des plus étranges" (1998, p. 7) et "qu'expliquer à un client qu'il rédige des exigences fondamentalement impossibles est embarrassant et n'est pas l'usage dans ces métiers" (idem, p. 6). Pierre Lévy souligne que "non seulement le but à atteindre n'est pas parfaitement défini

¹⁰⁴ Symétriquement, mais moins fréquemment, des demandes peuvent ne pas être formulées par des utilisateurs soucieux des coûts et des délais de réalisation de l'application, alors que leur satisfaction demanderait peu d'efforts de développement.

au départ, mais [qu'] il subit en cours de route nombre de remaniements essentiels" (1992, p. 44)¹⁰⁵.

Indépendamment de l'évolution des besoins pendant le processus de développement du logiciel, c'est surtout pendant sa phase d'utilisation que les besoins des utilisateurs se modifient. Ces besoins évoluent en fonction des nouvelles possibilités qu'apportent les évolutions rapides du matériel. Dans le domaine des logiciels, un produit paraît obsolète dès son achèvement, voire même avant (Frederick P. Brooks, 1996, p. 8). Certes, cette obsolescence relative concerne plus souvent des innovations qui sont plus au stade de l'idée nouvelle que de la réalisation effective, mais les demandes de modifications déstabilisent des applications qui avaient été implantées avec difficulté (Jean-Marie Desaintquentin, Bernard Sauteur, 1991, p. 17). Evidemment, les demandes de modifications existent également pour des produits matériels, "mais l'existence même d'un objet tangible sert à canaliser et à quantifier les demandes de changement de l'utilisateur, tandis que la souplesse et l'invisibilité d'un produit logiciel exposent ses constructeurs à d'incessants changements d'exigences" (Frederick P. Brooks, 1996, p. 99). Enfin au fur et à mesure de l'utilisation d'un logiciel, les caractéristiques auxquelles l'utilisateur accorde le plus d'importance évoluent : au départ l'utilisateur peut être particulièrement sensible à la simplicité d'utilisation d'un nouveau logiciel, alors qu'une fois la période d'apprentissage réalisée ce sont l'efficacité du produit et la richesse de ces fonctionnalités qui peuvent lui sembler décisives. Le logiciel ne peut s'adapter aux évolutions des compétences de ses utilisateurs : un logiciel qui explique dans le détail les opérations à effectuer peut-être plus facile à appréhender mais devenir rapidement fastidieux à utiliser.

Des caractéristiques contradictoires

Le problème est que les différentes caractéristiques d'un logiciel sont relativement contradictoires : par exemple, un logiciel écrit en langage Java est doté d'une très grande portabilité¹⁰⁶ mais est estimé deux à vingt fois moins rapide à l'exécution qu'un logiciel écrit

¹⁰⁵ Capers Jones, à partir de l'étude des devis de 64 projets effectués par IBM au début des années soixante-dix, constate que 35 % des fonctionnalités réalisées ne figuraient pas dans les spécifications initiales mais avaient été ajoutées après coup sur demande des clients (1989, p. 261-262).

¹⁰⁶ Il peut s'exécuter sur n'importe quelle machine disposant d'un petit programme d'interprétation de ce langage (machine virtuelle Java), ce qui explique son utilisation pour des applications stockées sur des serveurs et exécutées sur l'ordinateur de l'utilisateur lors d'une connexion Internet.

en C++ (Eurostaf, 1997 A, p. 26). Les compromis nécessaires entre les différents facteurs de qualité sont d'autant plus difficiles à réaliser que le logiciel est destiné à des utilisateurs hétérogènes. Il est en effet extrêmement rare qu'un logiciel soit développé pour un seul utilisateur. Les applications sur mesure sont destinées le plus souvent à des utilisateurs multiples et les progiciels sont par définition des produits standards destinés à des utilisateurs anonymes. Dans ce cas, le producteur de logiciel répond à un besoin "abstrait", c'est à dire le "besoin du marché dans son ensemble" (Faïz Gallouj, 1994, p. 195). L'utilisateur est un être abstrait qui recouvre une grande diversité d'utilisateurs concrets, alors même que le programmeur doit "inscrire dans un scénario fixé à l'avance les acteurs qui vont assumer le rôle de l'utilisateur, car l'usage d'un logiciel ne peut se dissocier d'une série de prescriptions plus ou moins naturelles, conviviales ou ergonomiques, mais toujours contraignantes" (Pierre Lévy, 1992, p. 23). "Un produit logiciel propre et élégant doit présenter à *chaque utilisateur* un modèle cohérent de l'application, des stratégies employées pour effectuer la tâche applicative, et des tactiques à appliquer pour spécifier les actions et leurs paramètres à l'aide de l'interface utilisateur" (Frederick P. Brooks, 1996, p. 223). Le problème est que "pour un produit logiciel donné, chaque attribut de l'ensemble des utilisateurs suit en fait une distribution, ayant de nombreuses valeurs différentes, chaque valeur ayant sa propre fréquence" (idem, p. 226). Dans ces conditions, il est tentant de surcharger le logiciel de fonctionnalités diverses (ce qui explique que ces fonctionnalités soient en moyenne faiblement utilisées), mais ce qui risque de nuire à l'efficacité du logiciel et à sa simplicité d'utilisation¹⁰⁷.

Une autre explication de l'insatisfaction des utilisateurs est que dans de nombreux cas ce ne sont pas eux qui sont les clients (acheteurs des produits ou donneurs d'ordres) des producteurs de logiciels. Cette situation d'une relation triangulaire entre producteurs, clients et utilisateurs n'est évidemment pas spécifique aux logiciels, mais elle est plus fréquente pour les produits informatiques, les services informatiques des organisations ayant souvent acquis un pouvoir de décision particulièrement important. On peut ajouter que les intérêts et les critères

¹⁰⁷ Roberto Di Cosmo et Dominique Nora (1998, p. 54) parlent "d'obésiciels" pour désigner les logiciels produits par Microsoft, qui au fil des versions successives se sont alourdis de multiples fonctionnalités peu utilisées. Microsoft fait toutefois remarquer que ces progiciels étant utilisés par des millions de personnes différentes, si les utilisateurs n'utilisent effectivement qu'une faible partie des possibilités, celle-ci est différente selon les utilisateurs. Ajoutons que l'importance des économies d'échelle fait qu'il est beaucoup plus rentable (du moins pour le moment) de produire un seul progiciel nécessairement sous-utilisé par grand type de besoin, que de le décliner en des versions différentes adaptées à des segments d'utilisateurs.

de choix de ces services sont certainement plus divergents de ceux des utilisateurs pour les produits informatiques, que pour les autres produits et services consommés par une organisation.

Les logiciels, boucs émissaires des mécontentements ?

Enfin, il faut prendre en compte le fait qu'une partie des reproches formulés par les utilisateurs aux logiciels le sont à tort. Il s'agit tout d'abord des cas où les dysfonctionnements observés proviennent en fait des compétences et de la formation insuffisantes des utilisateurs. Nous avons déjà signalé l'importance du rôle des futurs utilisateurs dans le développement des logiciels sur mesure, où on peut considérer qu'existe un processus de co-production, mais le savoir-faire et le savoir-utiliser des utilisateurs déterminent également la plus ou moins importante des potentialités du logiciel qu'ils pourront exploiter. Il faut notamment prendre en compte les exigences particulières à un traitement automatique, d'une extrême formalisation de l'ensemble des données introduites lors du processus d'utilisation. Le processus d'apprentissage, qui dépasse la seule dimension technologique, est plus délicat à mettre en œuvre quand l'utilisation du logiciel est imposée aux utilisateurs, que cette imposition soit le fait de la direction d'une entreprise vis à vis de ses salariés, ou d'une entreprise à ses sous-traitants¹⁰⁸. Jacques Printz estime qu'il faudrait pouvoir calculer "un indice de satisfaction qui fasse la part des choses entre ce qui est la responsabilité de l'utilisateur et celle du fournisseur de système", ce qui n'est pas simple à réaliser car il faut pouvoir "observer le système en des points qui ont à la fois un sens pour l'utilisateur (où tout est sémantique et pragmatique) et pour le concepteur de système (où tout est structure et syntaxe)" (1998, p. 288-289).

Dans un certain nombre de cas, l'insatisfaction concerne en réalité les réorganisations de l'activité opérées à l'occasion de l'introduction d'un logiciel, le logiciel ne faisant que "traduire" cette réorganisation, qui est en quelque sorte "neutralisée"¹⁰⁹. En effet,

¹⁰⁸ Ramon Salvador Valles, Lucas Van Wunnik, et Félix Pineda, montrent à propos des effets de l'échange de données informatisées (EDI) dans le secteur automobile, que quand cette introduction est imposée par de grandes entreprises aux PME sous-traitantes, l'apprentissage se limite à la maîtrise technologique des outils et n'intègre pas la composante organisationnelle de l'apprentissage, ce qui limite l'obtention des bénéfices potentiels de cette technologie (1997, p. 140).

¹⁰⁹ Ce phénomène s'apparente aux actions de résistance par rapport à l'introduction des machines, quand les salariés les jugent responsables de leur déqualification. Comme le fait observer, Harry M. Collins (1992), "ce ne sont pas les machines qui causent la déqualification, mais l'organisation du travail" (p. 156). Pour que la tâche effectuée par un opérateur humain puisse être réalisée par une machine, il faut "d'abord redéfinir le travail de telle sorte que celui-ci puisse être effectué de façon machinique" (en imitant les machines dans nos actes), et "c'est là qu'intervient le processus de déqualification" (idem, p. 289).

l'informatisation peut permettre de faire accepter des bouleversements organisationnels, qui n'auraient pas été acceptés en son absence (François Pichault, 1990, p. 99). Il faut également prendre en compte le fait que la modélisation systématique des organisations existantes effectuée lors de la conception d'un logiciel conduit à repérer les nombreuses incohérences existantes, qui résultent de la construction incrémentielle, par strates des procédures organisationnelles (Eric Brousseau, 1997, p. 54). La résorption de certaines failles organisationnelles peut conduire à en révéler de nouvelles. Des inefficacités organisationnelles, comme les redondances, peuvent être mises en évidence lors de la création d'un logiciel (Philip E. Agre, 1997, p. 251). La systématisation de la production, de la collecte et du stockage de l'information permet de procéder à des analyses de performances qui peuvent révéler des dysfonctionnements (Eric Brousseau, 1997, p. 56). "Les problèmes d'organisation et de management ainsi révélés sont réels et ne sont pas provoqués par l'informatisation", mais "l'informatique contribue cependant à les mettre en pleine lumière" (Jean-Marie Desaintquentin, Bernard Sauteur, 1991, p. 15). Un problème particulier est le dévoilement de la non-observation en pratique de certaines règles qui devaient être théoriquement appliquées¹¹⁰, les contraintes du traitement automatique nécessitant l'exposition systématique des pratiques effectives¹¹¹. La conception du logiciel nécessite l'élaboration d'un nouveau système de règles "plus compliqué (...) correspondant à la fois aux anciennes règles, aux pratiques des personnes, aux cas traités à part par la hiérarchie, aux spécificités des ordinateurs, à des souhaits d'évolution du travail, au traitement des erreurs" (Jean-Louis Peaucelle, 1997, p. 29). Ce nouveau système de règles élimine au maximum les "zones d'ombre", dont la sociologie des organisations de Michel Crozier et Ehrard Friedberg (1987) a pourtant mis en évidence l'importance pour que l'organisation puisse fonctionner. Cette théorie, qui analyse le pouvoir des acteurs comme la résultante de l'existence de "zones d'incertitudes", permet de comprendre les oppositions des utilisateurs à une véritable transparence organisationnelle, qui renforce les pouvoirs de la direction générale et/ou des

¹¹⁰ L'écart entre les pratiques effectives et les règles théoriques peut s'expliquer par une volonté des acteurs d'améliorer leur situation au détriment de l'organisation, mais également par le fait que le fonctionnement effectif de toute organisation est tributaire d'une interprétation des règles avec une certaine souplesse, ce que démontre a contrario l'efficacité des "grèves du zèle" qui consistent à appliquer scrupuleusement les règles existantes.

¹¹¹ Un rapport de l'Inspection Générale de l'Administration de l'Education Nationale consacré notamment à l'introduction des logiciels Nabucco (pour la comptabilité) et Apogée (pour la gestion des étudiants) souligne les très grandes libertés que les Universités prenaient avec les textes légaux et qui ont été révélées lors de l'implantation de ces logiciels. Le rapport précise en conclusion que "l'informatisation devrait aider les universités à s'inscrire dans des cadres plus respectueux des règles" (IGAEN, 1998, p. 54).

services informatiques à leur détriment. A partir de ce cadre d'analyse appliqué à l'étude de projets d'informatisation, Francis Pavé (1989) conclut que les projets réussis sont en fait les projets "dévoqués" dans leur utilisation (utilisés dans des perspectives non projetées au départ), car ce sont ceux qu'ont pu se réappropriés les utilisateurs, "les pratiques sociales [réussissant] à contourner la transparence organisationnelle" (p. 13).

En appliquant l'analyse de Jacques Girin, on peut dire que la qualité d'un logiciel définie comme étant "sa capacité à réaliser les performances que l'on attend de lui", est la "compétence d'un agencement" entre des "ressources humaines, matérielles et symboliques" (on peut ajouter informationnelles). Cette compétence "réside dans les relations entre ses divers éléments, bien plus que dans les propriétés de chaque élément pris séparément" (Jacques Girin, 1994, p. 27).

3 - La qualité des logiciels, une construction sociale conventionnelle

La qualité d'un logiciel est donc plus une construction sociale faisant intervenir des attentes réciproques entre les producteurs et les utilisateurs, qu'un ensemble de caractéristiques objectives universellement et atemporellement acceptées. Ce constat s'appuie sur l'analyse effectuée par Pierre-Yves Gomez (1994) de l'économie de la qualité. Cette analyse critique la vision de la qualité de la micro-économie standard, selon laquelle la qualité est un simple problème de différenciation des produits : dans un premier temps chaque objet à échanger possède un ensemble de caractéristiques qualitatives qui définissent le champ de leur utilité ; dans un second temps, lorsque la qualité est définie, le jeu classique offre/demande sur les prix (Pierre-Yves Gomez, 1994, p. 57-58). Il reste en effet à expliquer pourquoi et comment telle norme de qualité émerge de préférence à une autre. Le libre jeu du marché, basé sur l'axiome du choix autonome d'individus calculateurs, ne peut suffire à révéler le choix des acteurs, comme l'ont montré différents travaux : dans des situations d'asymétrie d'informations (imperfection de l'information sur le marché avec possibilité d'une connaissance cachée ou *sélection adverse*¹¹²), la qualité la plus basse peut l'emporter sur la qualité la plus élevée, ce que démontre George A. Akerlof (1970) à partir du marché des voitures d'occasions ; le prix n'est pas systématiquement un bon indicateur de l'information

¹¹² La sélection adverse (ou anti-sélection) est une situation où un des contractants dispose d'informations pertinentes pour la relation et ignorées de l'autre contractant, ce qui lui permet de bénéficier de rentes informationnelles.

sur les objets échangés (Joseph Stiglitz, 1985), il est plutôt "un signal qui suppose un accord préalable sur la qualité des objets" (Pierre-Yves Gomez, 1994, p. 62), d'où la nécessité d'une répétition des relations entre les acteurs, pour créer la confiance indispensable à l'émergence d'évaluations communes ; l'existence d'une garantie ne suffit pas à résoudre le problème de l'incertitude initiale sur la qualité, car la garantie constitue "un coût et devient donc elle-même un objet de calcul" source de nouvelles incertitudes avec des risques d'action cachée ou *moral hazard* (idem, p. 63-65) ; en raison de comportements opportunistes¹¹³ (par exemple incursions courtes sur des marchés), la libre concurrence peut être source de mauvaise qualité ; même en l'absence de comportements opportunistes, le théorème d'Arrow, qui s'inscrit dans la logique du paradoxe de Condorcet, montre l'impossibilité d'une émergence spontanée de la qualité, fondée sur la totale subjectivité des choix et le libre jeu du marché (idem, p. 67).

La nécessité d'intégrer la confiance et la convergence des représentations sociales implique de considérer la qualité comme une construction sociale, qui n'est pas la conséquence des rapports marchands, mais qui préexiste à l'échange et à la production (Pierre-Yves Gomez, 1994, p. 137). Pour l'étudier, il faut utiliser des "outils d'analyse alternative à l'axiomatique standard en puisant dans le corpus conceptuel produit par la théorie des conventions" (idem, p. 75)¹¹⁴. Les conventions ne portent pas directement sur la qualité des objets échangés mais sur le rôle des acteurs de l'échange, le type de comportement attendu de l'offreur et du demandeur. La qualité est définie par cet auteur comme une conjonction entre une convention d'effort et une convention de qualification. La convention de qualification "établit la compétence d'un professionnel", "elle offre une procédure de résolution récurrente de problèmes de détermination de la qualité lors de l'échange, en émettant une information sur les pouvoirs de qualifier qu'il s'agit d'attendre des professionnels d'une part, des clients de l'autre" (idem, p. 145) ; la convention de qualification établit un consensus entre le client et son fournisseur tel que l'un et l'autre savent ce qu'il faut attendre d'une relation marchande, elle opère dans le corps social le partage des compétences entre clients et professionnels. La convention d'effort supporte la réalisation de la qualité et se situe au niveau interne de la firme, la qualité résultant d'un effort (conventionnel) de production. De cette analyse de la

¹¹³ L'opportunisme est une forme de rationalité qui pousse des agents économiques à ne pas respecter un contrat s'ils n'y sont pas contraints et si cela leur permet d'améliorer leur situation personnelle. La sélection adverse et le risque moral sont deux types d'opportunisme.

¹¹⁴ Nous reviendrons sur le cadre théorique de l'économie des conventions dans le chapitre VII.

qualité, comme doublement contrainte par une convention d'effort et par une convention de qualification, découle logiquement qu'une diminution du niveau de qualité perçue provient des absences de cohérence, soit dans l'une des deux conventions, soit entre les conventions (idem, p. 217).

Cette analyse peut permettre de comprendre à la fois les tentatives de résoudre le problème de la qualité des logiciels par la certification des entreprises (notamment la norme ISO 9000-3 spécifique au logiciel) et l'échec relatif de ces tentatives. En effet, le présupposé de cette norme est qu'il suffit d'un bon processus de développement pour obtenir un "bon" produit et que la qualité du processus de développement est obtenue par une extrême formalisation des procédures mises en œuvre. Comme le souligne, Maarten Boasson, "le développement d'un système complexe a davantage de ressemblance avec la création d'une œuvre d'art qu'avec la simple production, et (...) il n'existe aucune relation démontrable entre le processus de développement et la qualité du produit résultant. L'essence du développement est qu'il y a des compromis à faire et des décisions à prendre qui exigent connaissances, compréhension et, ce qui est malheureusement trop souvent contesté, intuition. L'idée que ces facteurs essentiels pour la qualité du produit final, puisse être éliminés en décrivant minutieusement dans le détail comment le processus de développement doit se dérouler est une erreur de raisonnement fondamentale" (1998, p. 4-5)¹¹⁵. Une démarche de type ISO 9000 ne peut répondre que très partiellement aux exigences de qualité des utilisateurs. C'est ce que permet de mettre en évidence l'analyse de Pierre-Yves Gomez qui montre non seulement pourquoi et comment la qualité évolue dans le temps (passage de la "qualité inspection" et de la "qualité contrôle" caractéristiques du taylorisme, à la "qualité assurance" du fordisme et émergence actuellement d'une "qualité totale"), mais également pourquoi des formes de qualités distinctes peuvent coexister, en conséquence de la pluralité des cohérences possibles entre des conventions différentes d'effort et de qualification.

Il reste à expliquer dans le cadre de l'économie des logiciels, pourquoi ces cohérences ont tant de difficultés à émerger. Une première réponse possible a trait aux relations initialement très déséquilibrées entre les informaticiens et les utilisateurs.

¹¹⁵ Pour un bilan très critique des normes ISO 9000 concernant les logiciels, cf. également Jacques Printz (1998, p. 289-290).

B - UNE RELATION INITIALE DESEQUILIBREE ENTRE INFORMATIENS ET UTILISATEURS

A l'origine les producteurs de logiciels ont été dans une situation de domination forte des utilisateurs, ce qui a rendu difficile la stabilisation de conventions de qualité.

Cette domination se situait tout d'abord sur un plan technique. Elle était particulièrement présente au début de l'informatique, dans la mesure où les possibilités relativement limitées des premiers ordinateurs et l'absence de programmes utilitaires bien conçus pour gérer ces ressources, faisaient que l'essentiel des efforts des informaticiens portaient sur les aspects les plus proches du matériel au détriment des aspects les plus proches des utilisateurs (Jean-Marie Desaintquentin, Bernard Sauter, 1991, p. 6). L'intercompréhension était particulièrement difficile, jusqu'au niveau du langage utilisé, entre des informaticiens connaissant mal les caractéristiques spécifiques des besoins des utilisateurs, et des utilisateurs peu au fait des contraintes particulières d'un traitement automatique : "les clients pensent en termes de flux d'air ou de sommes à facturer, alors que les développeurs réfléchissent en termes de structures de données et d'interfaces entre modules. Il peut arriver que le même terme -fichier par exemple- ait un sens différent pour les uns et pour les autres" (Lauren Ruth Wiener, 1994, p. 115). La relation qui s'instaurait entre eux était moins une relation de coproduction, qu'une relation maître-élève. Au nom d'exigences d'ordre technique, les utilisateurs étaient contraints d'accepter les logiciels tels qu'ils avaient été conçus par les informaticiens et de s'y adapter. Cette situation s'est pérennisée en raison de l'évolution accélérée des techniques, sans qu'il y ait en même temps de "véritable progression de la maîtrise technologique" (Serge Bouchy, 1994, p. 172).

Ces impératifs techniques sont renforcés par la culture spécifique des informaticiens : "beaucoup d'informaticiens de métier aiment suffisamment l'informatique pour construire d'abord des programmes qui sont un prétexte à une relation de qualité avec l'ordinateur plutôt que des outils véritablement utilisables par l'environnement. Quelle entreprise ne résonne pas des plaintes de ceux à qui l'on propose des programmes parfaitement "esthétiques" d'un point de vue informatique, mais assez peu conforme au simple "outil" qu'attendent les utilisateurs" (Philippe Breton, 1990, p. 46). Les informaticiens accordent souvent plus d'importance aux caractéristiques techniques des logiciels produits qu'à leurs caractéristiques d'usage. Jacques Printz estime que "les programmeurs ont souvent tendance à fabriquer de la complexité sans véritable justification économique, car c'est une façon de montrer leur maîtrise logique"

(1998, p. 83). Philippe Breton considère même "qu'aucun développeur de logiciels qui se respecte, ne peut construire un programme qui ne fasse appel à un minimum "d'esprit informatique" de la part de celui qui l'utilise" (1990, p. 85)¹¹⁶. "Jusqu'à une période récente, la pénurie apparente en matière de compétences informatiques d'une part, et le côté pionnier de cette activité d'autre part" (Serge Bouchy, 1994, p. 174) faisaient accepter cette situation.

Sur le plan interne aux organisations, les informaticiens ont un pouvoir important. Dans leur capacité à "débuguer" les programmes, ce pouvoir s'apparente à celui des ouvriers d'entretien analysé par Michel Crozier dans le monopole industriel. Le pouvoir des informaticiens est à son apogée lors de la première période de l'informatique, caractérisée par de grands systèmes centralisés, selon le modèle qu'IBM a développé chez ces clients (Claire Charbit, Jean-Benoît Zimmermann, 1997, p. 6). L'introduction de ces systèmes implique fréquemment une refonte des structures organisationnelles, dans laquelle l'avis des informaticiens est déterminant¹¹⁷. Initialement techniques, les compétences des informaticiens deviennent sociales, les utilisateurs devant négocier avec eux la forme et souvent le contenu de chaque information qui circule dans l'entreprise (Philippe Breton, 1987, p. 208). Dans le même temps, les utilisateurs voient leur propre pouvoir diminuer, en raison de la dépossession des informations qui résulte de leur centralisation, ce que Jean-Marie Desaintquentin et Bernard Sauteur appelle un "hold-up sur les fichiers" (1991, p. 10). Les nouveaux services informatiques deviennent de plus en plus proche de la direction générale de l'entreprise, qui se contente le plus souvent d'entériner leurs choix, en étant incapable de les apprécier par elle-même (idem, p. 17). Ce règne des informaticiens correspond toutefois moins à une volonté de prise de pouvoir de salariés experts, qu'aux conséquences du désir des directions - très empreint de rationalité technicienne - de transférer les mécanismes décisionnels aux machines (Philippe Breton, 1987, p. 207)¹¹⁸.

¹¹⁶ Cette conception élitiste de l'utilisation de l'informatique est illustrée par le véritable mépris de la communauté informatique dans les années soixante et soixante-dix, pour les informaticiens qui travaillaient sur les interfaces hommes-machines, selon Nicholas Negroponte (1995, p. 116).

¹¹⁷ Jean-Marie Desaintquentin et Bernard Sauteur soulignent le rôle de ces ingénieurs technico-commerciaux, délégués par les constructeurs de matériel informatique, qui ne connaissaient rien aux métiers de la banque, de l'assurance ou de l'industrie, et qui se retrouvaient installés aux commandes du système d'information de ces entreprises (1991, p. 6).

¹¹⁸ De même, Harry M. Collins constate "l'aspect séduisant qu'ont en commun la bureaucratie et la tyrannie des machines : (...) tous les doutes, et la responsabilité qui y est liée de devoir prendre des décisions dans des conditions d'incertitude, ont été levés" (1992, p. 202).

L'externalisation éventuelle du développement des applications ne change pas significativement la situation, du moins durant les premières périodes de l'informatique. En effet, globalement le marché est dominé par l'offre. L'existence de systèmes "propriétaires" limite singulièrement la concurrence et place les clients dans une situation de dépendance vis à vis des fournisseurs (*customer lock-in*), avec des coûts de transfert très élevés pour une activité qui acquiert un caractère de plus en plus stratégique. L'asymétrie d'informations entre le mandant et le mandataire, liée à la faible substituabilité entre fournisseurs, augmentent les risques de comportements opportunistes tirant partie des failles de contrats nécessairement incomplets (Jacques Girin, 1994, p. 10). De plus les relations avec les fournisseurs externes restent le plus souvent l'apanage des services informatiques internes, qui peuvent se sentir plus solidaires de ces fournisseurs, dont ils partagent la culture, que de leur propre organisation (idem, p. 24).

Toutefois les relations entre les informaticiens et les utilisateurs vont progressivement se rééquilibrer sous l'influence de plusieurs facteurs. Tout d'abord la substitution d'architectures ouvertes aux systèmes "propriétaires" introduit une plus grande concurrence, sur des marchés moins dominés par l'offre et qui ont déjà connu une période de récession. Ensuite et surtout, l'irruption de la micro-informatique introduit de profonds bouleversements qui vont avoir des conséquences sur l'ensemble de l'informatique. Une part croissante des dépenses informatiques est effectuée directement par les services utilisateurs¹¹⁹. La diffusion des micro-ordinateurs permet une démystification et une plus grande appropriation sociale de l'informatique par les utilisateurs, qui se montrent beaucoup plus exigeants, notamment en termes de convivialité des applications utilisées, ce que facilite la formidable augmentation des capacités de traitement et de mémorisation du matériel. En même temps, l'augmentation des compétences des utilisateurs les rend plus sensibles aux exigences particulières d'un traitement automatique, et plus réalistes dans la formulation de leurs besoins. Certes, plus récemment la mise en réseau progressive de l'ensemble des ressources informatiques des organisations permet une certaine reprise en main par les services informatiques, pour garantir la cohérence globale du système d'information, mais celle-ci doit bien d'avantage tenir compte des souhaits d'utilisateurs qui sont devenus plus autonomes. "L'époque où l'esthétique

¹¹⁹ En Europe, la part des dépenses informatiques effectuées par les directions informatiques passe de 70 % en 1993 à 58,3 % en 1995, suivant la tendance qu'avaient connu les Etats-Unis (Eurostat, 1996 A, p. 210).

technique d'une solution suffisait à la justifier est définitivement révolue" (Jean-Marie Desaintquentin, Bernard Sauteur, 1991, p. 56).

C - DES PROGRES REELS MAIS UNE INSATISFACTION MAINTENUE DES UTILISATEURS

Le rééquilibrage des relations entre informaticiens et utilisateurs est une des causes des améliorations qualitatives incontestables des logiciels (1). Toutefois, la diversité croissante des types de logiciels, de leurs utilisateurs et les possibilités de comportements opportunistes rendent difficiles la stabilisation de conventions de qualité (2).

1 - Amélioration de la "convivialité" et nouvelles méthodes de développement

Ces améliorations qualitatives concernent principalement l'augmentation de la "convivialité" des applications et une meilleure perception des besoins des utilisateurs grâce à l'apparition de nouvelles méthodes de développement.

Un outil peut être défini comme convivial, lorsqu'il est mis "au service de la personne intégrée à la collectivité et non au service d'un corps de spécialistes" (Ivan Illich, 1973, p. 13). Une première amélioration importante résulte du passage du modèle "batch" au modèle "conversationnel", devenu le modèle dominant avec l'apparition des terminaux "intelligents", puis des micro-ordinateurs, et le développement des architectures clients-serveurs. Dans le modèle "batch" (ou traitements par lots), l'utilisateur soumet à l'ordinateur un ensemble de travaux à effectuer, les données résultant des traitements effectués par l'ordinateur lui parvenant globalement quelque temps après, sans que l'utilisateur ait la possibilité d'intervenir pendant les traitements. Par contre dans le modèle "conversationnel" (temps réel et transactionnel), les interactions sont permanentes entre l'ordinateur et l'utilisateur, celui-ci pouvant ajuster continuellement les demandes effectuées à l'ordinateur au vu des résultats délivrés quasi-instantanément par la machine. Il faut souligner que la plus grande souplesse qu'apporte ce mode de traitement à l'utilisateur, a nécessité une importante augmentation de la complexité des logiciels pour maintenir la fiabilité et la performance des traitements, et l'intégrité des données (Jacques Printz, 1998, p. 198-200).

La deuxième amélioration, qui complète la précédente, est l'amélioration des interfaces entre l'utilisateur et l'ordinateur, avec l'apparition de l'interface graphique. Les interfaces graphiques résultent des travaux effectués par Xerox au PARC, et ont été intégrées dans un

produit commercialisé par Apple avec le MacIntosh dans les années quatre-vingt, avant de se généraliser à l'ensemble des ordinateurs. Elles reposent principalement sur la substitution d'un écran graphique (écran *bit-map*) au traditionnel écran alphanumérique et l'utilisation d'un dispositif de pointage (principalement la souris), qui facilitent considérablement les communications avec le système informatique par l'utilisation de symboles graphiques (icônes), des menus déroulants, et par le réalisme des représentations dans des "fenêtres" présentes sur un "bureau". Il est significatif, de la conception des informaticiens des compétences que devait posséder un utilisateur, que la souris et les fenêtres furent originellement conçues pour les handicapés mentaux incapables de verbaliser leurs requêtes (Emmanuel Saint-James, 1993, p. 8). Une illustration des possibilités que pouvaient apporter des logiciels de ce type, est le succès du tableur, qui fut un des éléments déterminants dans l'avènement de la micro-informatique. Le tableur n'oblige pas l'utilisateur "à s'engager dans des processus incompréhensibles mais va droit au but et leur donne un accès immédiat à la puissance des machines à partir d'une formulation de leurs problèmes en termes simples : les termes mêmes qu'ils employaient avant que l'outil informatique leur soit proposé" (Jean-Marie Desaintquentin, Bernard Sauteur, 1991, p. 25). Cependant les améliorations ergonomiques des logiciels, qui nécessitent la consommation de ressources matérielles plus abondantes et moins coûteuses, entraînent également une augmentation de la complexité des logiciels (programmation "événementielle").

Un autre domaine qui a permis des améliorations qualitatives des logiciels est l'apparition de nouvelles méthodes de développement. Les méthodes traditionnelles basées sur une première phase d'écriture de spécifications détaillées devant rester intangibles durant tout le processus de développement du logiciel, processus qui peut être long, se sont révélées relativement inadaptées pour appréhender les besoins réels des utilisateurs, tels qu'ils s'expriment lors de l'emploi du logiciel. Effectuer des revues et relectures de documents volumineux et rébarbatifs pour considérer que les spécifications d'un logiciel sont acceptées par les utilisateurs, est insuffisant. En effet, d'une part les besoins évoluent, et d'autre part, il est souvent difficile pour des futurs utilisateurs, dont la motivation peut être variable, d'exprimer abstraitement leurs besoins¹²⁰. Le prototypage rapide, les méthodes de

¹²⁰ Par exemple, en étudiant le développement d'un grand système d'informations (4000 hommes*mois), le projet SIMAT (Système d'Informations de la Maintenance de l'Armée de Terre), Jean-Paul Hamon constate que les besoins exprimés en phase de "définition des besoins" sont entachés de 30 % d'incertitude (1996, p. 18).

développement incrémentales et la conception participative¹²¹, qui donnent la possibilité aux utilisateurs d'apprécier concrètement les futures caractéristiques du logiciel et de réagir durant le processus de développement, peuvent permettre de mieux répondre aux besoins réels des utilisateurs. Les technologies objet, notamment dans les possibilités de constituer des "objets-métiers" caractéristiques des activités des utilisateurs, peuvent faciliter le dialogue et l'intercompréhension entre concepteurs et utilisateurs des applications¹²².

2 - Les difficultés pour stabiliser des conventions de qualité

Malgré ces progrès qualitatifs indéniables, il existe une insatisfaction maintenue des utilisateurs. Une des explications de ce paradoxe nous semble résider dans les difficultés pour stabiliser des conventions de qualité. On peut ajouter que dans certains cas, les améliorations observées rendent même encore plus difficile l'émergence de telles conventions.

Tout d'abord il faut prendre en compte l'élargissement continu et la diversité croissante des utilisateurs de logiciels. En même temps que les compétences des anciens utilisateurs augmentent en raison des phénomènes d'apprentissage par l'usage (*learning by using*), apparaissent de nouveaux utilisateurs inexpérimentés. Quand c'est le même logiciel qui est destiné à être employé par des utilisateurs très hétérogènes (grand système impliquant des milliers d'utilisateurs ou progiciel vendu à d'innombrables clients), il devient difficile de réaliser un produit capable de répondre simultanément à des exigences différentes. Paradoxalement, l'amélioration des interfaces entre l'ordinateur et l'utilisateur accentue ce problème, dans la mesure où la conception de telles interfaces nécessite des présuppositions fortes sur le comportement de l'utilisateur, en raison notamment de l'importance de ce qui est considéré comme implicite par le programme lors d'une action explicite de l'utilisateur. En effet, demander à l'utilisateur d'explicitier l'ensemble des paramètres de sa requête rend celle-ci rapidement très laborieuse à construire. Une autre conséquence de l'amélioration des

¹²¹ Ces méthodes ont été présentées dans le chapitre III.

¹²² Par exemple la technique des cas d'utilisation, introduits par Ivar Jacobson (1993). Cette technique détermine les besoins fonctionnels selon le point de vue d'une catégorie d'utilisateurs à la fois. Facilement compréhensibles par les utilisateurs potentiels du système, les cas d'utilisation permettent de valider les fonctionnalités à l'aide de scénarios. La réalisation des cas d'utilisations est effectuée par une société d'objets collaborant, à partir des objets du domaine et de leurs relations.

interfaces est qu'elles peuvent contribuer à faire prêter inconsciemment à l'ordinateur des fonctions de bon sens dont il est parfaitement incapable (Jacques Printz, 1998, p. 39)¹²³.

La deuxième difficulté pour stabiliser des conventions de qualité réside dans l'apparition continue de logiciels répondant (au départ souvent imparfaitement) à de nouveaux besoins. Ces logiciels correspondent le plus souvent à des projets de plus en plus importants, qui ne peuvent être réalisés qu'en répartissant le travail entre des sous-équipes, avec le risque que chaque sous-équipe optimise mal sa partie pour atteindre ses objectifs, au détriment des qualités globales du produit (Frederick P. Brooks, 1996, p. 207). Enfin, pour un logiciel d'un type donné, de nouvelles versions sont mises sur le marché à un rythme extrêmement rapide, en raison des facilités de modification d'un produit existant sous forme numérique et des intérêts des producteurs. Ceci a pour conséquence une "obsolescence rapide des connaissances" des utilisateurs, qui "rend épuisante la poursuite d'un niveau élevé de maîtrise" (Jean-Pierre Faguer, Michel Gollac, 1997, p.115), l'apprentissage de l'utilisation d'une nouvelle version nécessitant des désapprentissages, source de confusions ou d'erreurs.

Ceci nous conduit à la troisième explication qui est la facilité pour les producteurs d'adopter des comportements opportunistes, concernant certaines dimensions qualitatives qui ne peuvent être perçues par les consommateurs, notamment *ex ante* lors de l'achat du produit. Il importe de distinguer de ce point de vue le cas des logiciels développés sur-mesure du cas des progiciels. Pour un logiciel sur mesure, le produit (et donc *a fortiori* ses caractéristiques qualitatives) n'existe pas au moment de la signature du contrat. Le contrat ne peut porter que sur des exigences, mais dont la "quantification est, dans la plupart des cas, soit difficile, soit invérifiable en pratique" (Philippe Robert, 1997, p. 169). De plus, "on ne peut pas démontrer, le plus souvent, que la satisfaction des exigences implique celle des besoins" (idem). Cette situation d'incomplétude des contrats est encore plus importante avec l'utilisation des nouvelles méthodes de développement (cf. supra) où les spécifications détaillées du logiciel sont élaborées progressivement lors du processus de développement du logiciel par des interactions continues entre les concepteurs et les utilisateurs. Si ces méthodes permettent de réduire l'écart entre les besoins latents des utilisateurs et les besoins perçus et exprimés par les

¹²³ Une étude fine des conséquences de la diversité des utilisateurs et de leurs compétences par rapport à l'utilisation d'un logiciel, un automate point de vente de billets SNCF, a été réalisée par Marc Breviglieri (1997). Sur l'absence de "bon sens" de tels programmes, on peut mentionner la recherche des trains en partance de Lille, depuis l'existence d'une seconde gare TGV distante d'une cinquantaine de mètres de la précédente. Dans certains cas, le programme ne propose que les trains en partance d'une des deux gares !

concepteurs, elles empêchent par définition l'inscription des spécifications précises dans le contrat initial. De ce fait, *une part importante des relations entre les producteurs de logiciels et leurs clients ne peut être basée que sur une confiance réciproque*. Celle-ci peut être particulièrement difficile à établir, et des incompréhensions successives peuvent rapidement détériorer les relations entre les partenaires comme l'atteste la fréquence des recours juridiques dans le domaine des logiciels sur mesure. La tentation peut être grande pour les producteurs de privilégier la productivité (permettant de maîtriser les délais et les coûts) au détriment de la qualité. Certes, certaines études estiment que des exigences accrues de qualité peuvent influencer positivement la productivité des développeurs en augmentant leur satisfaction personnelle et leur motivation¹²⁴, mais il ne peut s'agir que de la qualité telle qu'elle est perçue par les développeurs, dont on a vu qu'elle pouvait reposer sur des critères différents de ceux des utilisateurs.

La situation est différente pour les progiciels où le produit existe au moment de son achat par un client, même s'il ne lui est pas toujours possible de connaître ses qualités réelles avant de l'avoir utilisé, notamment en raison du renouvellement très rapide des produits. Surtout l'importance des rendements croissants d'adoption, et notamment des externalités de réseaux, permet d'expliquer comment des progiciels, indépendamment de leurs qualités peuvent s'imposer sur un marché¹²⁵. De ce fait les stratégies des producteurs de progiciels peuvent ne pas accorder à la qualité du produit l'importance souhaitée par les utilisateurs.

Indépendamment des comportements des producteurs, les difficultés pour répondre aux besoins des utilisateurs s'expliquent également par des caractéristiques spécifiques des logiciels, tant au niveau de leur conception que de leur utilisation.

¹²⁴ Toran Demarco, Timothy Lister estiment même que "des produits d'une qualité bien supérieure à celle qu'accepte l'utilisateur final sont synonymes de meilleure productivité" (1991, p. 31). En effet, pour un programmeur "la nécessité où il se trouve d'avoir à livrer un produit nettement inférieur à ce qu'il sait pouvoir réaliser est une atteinte à son amour-propre et compromet toute satisfaction personnelle" (idem, p. 146). Réaliser des produits de piètre qualité, pour répondre aux contraintes de délais et aux exigences immédiates des clients a des effets négatifs en augmentant l'insatisfaction des développeurs sur leur comportement et leur efficacité. Ces auteurs citent les exemples au Japon de certaines firmes (Hitachi Software, certains secteurs de Fujitsu) où "l'équipe des réalisateurs a le pouvoir effectif de s'opposer à la livraison d'un produit qu'elle estime ne pas être vraiment prêt" même si "le client est d'accord pour l'accepter en l'état" (idem, p. 33).

¹²⁵ Ces mécanismes seront décrits précisément dans le chapitre VII.

D - LES DIFFICULTES DUES AUX SPECIFICITES DU LOGICIEL

Au niveau de la conception, le problème provient de l'écart inévitable entre une modélisation (point de passage obligé pour concevoir un logiciel) et la partie de la réalité à modéliser (1). Au niveau de l'utilisation, il s'agit des problèmes de "communication" entre un logiciel et ses utilisateurs (2). Certes, ces problèmes existent pour beaucoup d'autres artefacts, dont la conception repose sur une modélisation et dont l'utilisation nécessite des interactions avec des êtres humains, mais les particularités du logiciel font qu'ils prennent dans ce cas une importance particulière.

1 - L'écart entre la modélisation et la réalité

La conception d'un logiciel passe nécessairement par une phase de modélisation¹²⁶ de la réalité concernée, qu'elle soit physique, sociale, psychologique ou institutionnelle. C'est à partir de cette modélisation que le problème à résoudre sera traduit en instructions exécutables par un ordinateur au cours du développement du logiciel. Une modélisation constitue une abstraction basée sur une rationalité de type logico-mathématique ; par définition, elle réalise une simplification de la réalité concrète, elle ne peut être qu'un décalque imparfait et incomplet de celle-ci.

Une modélisation destinée à être traduite en un programme informatique a des exigences particulières, quasiment contradictoires avec cette définition, dans la mesure où elle n'est pas simplement destinée à représenter un phénomène pour l'expliquer, mais à réaliser des traitements automatiques. Elle devrait donc être exhaustive (en n'omettant aucun élément pouvant être pertinent), totalement déterministe et ne tolérer aucune ambiguïté pouvant induire plus d'une interprétation possible. Or la réalité, notamment sociale, ne peut être connue parfaitement et complètement, et recèle toujours une part importante d'ambiguïté et d'interprétation pour que les organisations puissent fonctionner réellement¹²⁷. La modélisation

¹²⁶ Le terme de modélisation "recouvre un vaste champ de significations, depuis la mathématisation d'un processus jusqu'au choix des données susceptibles de constituer a priori toute l'information nécessaire et suffisante pour représenter l'activité d'une partie de l'entreprise" (Jean-Louis Peaucelle, 1997, p. 22).

¹²⁷ "L'indéterminisme, ou l'ambiguïté, est utile dans de nombreuses situations humaines et se prête très bien au flou de certaines situations sémantiques" (Jacques Printz, 1998, p. 345).

est donc nécessairement "imparfaite"¹²⁸ parce qu'il ne peut exister "un ensemble parfait de règles [qui] aurait à couvrir toutes les situations possibles (...) [et qui] constituerait une description complète du passé et de tous les futurs possibles" (Harry M. Collins, 1992, p. 130). Il faudrait, pour les règles gouvernant le comportement humain, pouvoir connaître "des règles prescrivant comment agir en toute circonstance concevable, sinon encore conçue, et de règles dans les règles pour leur dire ce qu'il faut rechercher pour identifier ces circonstances, et de règles dans les règles pour les aider à identifier les choses qui doivent les aider à identifier les circonstances, et ainsi de suite *ad infinitum*" (idem, p. 131).

Pour que le logiciel résultant de cette modélisation puisse néanmoins être opérationnel, il importe que les utilisateurs assimilent les inévitables simplifications qui ont été réalisées par les concepteurs, et que ces simplifications soient pertinentes par rapport à la réalité de l'activité informatisée. "Tout acte de programmation projette, en fait, à travers le programme dont il est le résultat, une certaine vision logique de la réalité telle qu'elle est perçue par le programmeur ou le concepteur" (Jacques Printz, 1998, p. 100). La qualité de la réponse apportée par le logiciel aux besoins des utilisateurs dépend donc de manière importante de la qualité de l'intercompréhension entre concepteurs et utilisateurs.

Des difficultés communicationnelles

Une première difficulté est d'ordre communicationnel. Nous avons déjà mentionné l'absence de solution satisfaisante pour énoncer les spécifications à la base de la modélisation : rédigées en langage naturel elles contiennent une foule d'ambiguïtés ; formulées dans un langage formel de spécifications elles sont incompréhensibles pour les utilisateurs et les experts du domaine¹²⁹. Plus généralement, Lauren Ruth Wiener note que "la communication qui se déroule entre clients et développeurs est profondément viciée", car "ces deux groupes de personnes vivent dans des mondes qui n'ont rien à voir" (1994, p. 115). Toran Demarco et Timothy Lister soulignent à juste titre que la "sphère d'activité [des concepteurs de logiciels] met en jeu la capacité à la communication inter-humaine plus que la capacité à communiquer

¹²⁸ "Enormément de projets échouent précisément à cause d'aspects qui n'avaient jamais été définis" (V.A. Vyssotsky, 1972).

¹²⁹ Lauren Ruth Wiener cite l'exemple de l'entreprise Telectronics Pacing Systems qui développe des logiciels pour des défibrillateurs cardiaques. Vu l'importance des problèmes de fiabilité, cette société entreprit de faire écrire les spécifications en langage formel, mais de ce fait les experts cardiologues ne furent plus en mesure d'affirmer si telle ou telle spécification devenue pour eux incompréhensible, était correcte ou totalement aberrante (1994, p. 144).

avec des machines" (1991, p. 112). Cependant, Jacques Printz remarque que "des personnalités à profil très technique comme les programmeurs ont souvent peu de talent, et peu de doigté en matière de communication" (1998, p. 83). La communication est d'autant plus difficile que "l'environnement dans lequel "fonctionne" un programmeur, et *a fortiori* un concepteur de systèmes informatiques est probablement parmi les plus compliqués que l'on puisse rencontrer dans les métiers d'ingénierie, complication aggravée par le côté non matériel et purement abstrait des entités manipulées qui fait que le non expert n'y comprend rien" (idem, p. 86). Dans toute communication, l'implicite occupe une place importante, notamment dans l'utilisation des métaphores et des analogies, rendant plus difficile la compréhension entre des membres de groupes ayant des références culturelles (techniques et professionnelles) différentes¹³⁰.

Des difficultés cognitives

A ces difficultés communicationnelles, s'ajoute l'écart entre ce que les individus disent ("*espoused theory*", les énoncés des acteurs sur leurs comportements) et ce que les individus font réellement ("*theory in use*", qui gouverne effectivement l'action des gens) (Chris Argyris et Donald A. Schön, 1978). Cet écart s'explique tout d'abord par l'importance, la diversité et la complexité des connaissances tacites (cf. chapitre I)¹³¹ : "il se peut que le savoir livresque soit plus aisément exprimable, précisément parce que sa nature abstraite lui confère une constante et inévitable absence de familiarité. Il nous est en permanence extérieur, ce qui en fait un élément visible en permanence de notre vie mentale" (Harry M. Collins, p. 202-203)¹³². Cet écart s'explique également par la distance existant entre les règles et les procédures formelles, et les routines réellement existantes (Jean-Claude Tarondeau, 1998, p. 43). Certes, "toute activité humaine est codifiée par des règles", mais "ces règles ne sont qu'en partie explicites, leur efficacité tient au fait qu'elles ne se dévoilent qu'en situation et qu'elles ne sont jamais

¹³⁰ Robert Michon et Lin Gingras (1988) parlent du "fossé sémantique" qui empêche une véritable compréhension mutuelle entre concepteur et utilisateurs des systèmes d'information.

¹³¹ "La règle générale est que nous en savons plus que ce que nous pouvons en dire et, si nous réussissons à en savoir plus que ce que nous pouvons en dire, c'est parce que nous apprenons par le biais de notre socialisation et non grâce à l'instruction" (Harry M. Collins, 1992, p. 22).

¹³² Capers Jones note que, y compris "dans les activités les plus simples du genre des programmes de saisie de commande ou de paie, il a été maintes fois remarqué que même des praticiens d'expérience, par exemple des personnes employées depuis plusieurs années à l'administration des ventes, ne savaient pas toujours expliquer ce qu'ils faisaient, en termes utilisables à la reprise par la machine des tâches humaines concernées" (1989, p. 133).

appliquées mais montrées, interprétées et éprouvées dans l'interaction et la négociation" (Michel Callon et Bruno Latour, 1991, p. 15). Pour gérer les événements et les aléas de façon efficiente, les opérateurs doivent mobiliser des capacités d'initiative et des savoirs diffus, souvent ignorés et occultés par les directions des organisations (Anne Mayère, 1997, p. 205-208). La modélisation ne peut s'appuyer sur aucune convention implicite (au sens d'écran informationnel, d'économie d'information) entre un système informatique et des êtres humains, et ces conventions sont limitées entre les concepteurs et les utilisateurs, alors même que les conventions les plus implicites - qui semblent "naturelles" - sont les plus susceptibles d'être omises dans la modélisation : les routines sont fréquemment d'autant plus efficaces que nous sommes moins conscients des connaissances sur lesquelles elles reposent et qu'elles s'apparentent à des réflexes. En effet, un apprentissage peut être analysé comme s'effectuant en trois étapes : lors de la première étape (étape cognitive), un individu apprend à partir d'instructions ou d'observations sur le type d'actes appropriés dans telles ou telles circonstances ; dans un second temps, ces apprentissages de la première phase sont mis en pratique jusqu'à ce qu'ils deviennent aisés et précis (étape associative) ; enfin, dans la phase autonome, les actions intériorisées par la pratique sont effectuées de moins en moins consciemment, avec une rapidité et une dextérité de plus en plus importantes, mais le savoir déclaratif s'étant transformé en une forme procédurale, il ne nous est plus accessible et nous ne pouvons plus l'exprimer verbalement (Harry M. Collins, 1992, p. 110). En conséquence, la connaissance des savoirs utiles, notamment pour transformer les heuristiques pratiquées en des algorithmes exécutables par un ordinateur¹³³, requiert fréquemment l'observation directe, voire l'apprentissage, par les concepteurs du comportement concret des utilisateurs. Par exemple, Ikujiro Nonaka et Hirotaka Takeuchi citent l'exemple de la conception d'un logiciel pour piloter une machine à fabriquer le pain ; les échecs pour formaliser correctement le pétrissage de la pâte ne purent être surmontés que par l'apprentissage par la responsable du développement du logiciel et plusieurs ingénieurs, de la fabrication du pain, auprès d'un boulanger célèbre pour la qualité de ses pains ; cet apprentissage direct fut indispensable pour réussir à "socialiser la connaissance tacite du maître boulanger par l'observation, l'imitation et la pratique" (1997, p. 85).

¹³³ Une heuristique donne des règles générales pour atteindre des objectifs généraux, alors qu'un algorithme indique la solution optimale pour atteindre un but bien défini.

Des difficultés socio-politiques

Les difficultés ne sont pas seulement d'ordre communicationnel ou cognitif, notamment lorsque l'introduction du logiciel a des conséquences organisationnelles. Le développement des logiciels est fréquemment basé sur une vision de l'organisation comme étant apolitique, neutre, où tous les acteurs poursuivent un objectif commun, et où l'exhortation et la pression hiérarchique suffisent pour s'assurer une collaboration active des utilisateurs. Si à l'inverse on envisage une organisation comme un lieu de conflit entre différents pouvoirs, dont la source réside notamment dans la détention de connaissances et le contrôle de l'information¹³⁴, on comprend que cette participation ne va pas nécessairement de soi. Une attitude de non-collaboration ne doit pas être analysée comme une peur irrationnelle du changement ou un simple "refus du changement", motivé par la nécessité d'effectuer de nouveaux apprentissages. Elle renvoie à un comportement rationnel basé sur la compréhension qu'au-delà de la représentation de l'activité actuelle, ce qui est en jeu est la réorganisation de l'activité, avec ses conséquences en termes de répartition des pouvoirs, de nouvelles tâches à effectuer, et, dans certains cas d'expropriation de connaissances détenues par les opérateurs.

Tout d'abord l'introduction d'un logiciel est l'occasion d'une restructuration des activités humaines, présentée comme n'étant pas inventée, mais simplement découverte, alors qu'elle résulte d'une construction qui nécessite une phase d'imposition importante (Philip E Agre, 1997, p. 251-258). Déjà en 1970, Michel Crozier soulignait que "l'application du système rationnel des informaticiens au système social que constitue une entreprise se heurte à des obstacles beaucoup plus profonds que ceux auxquels on pense généralement" dans la mesure où "il ne s'agit pas, seulement d'habitudes qu'il faudrait changer ni même d'intérêts qui seraient menacés" mais "de tout un ensemble de pratiques et d'arrangements qui constituent en fait le mode de gouvernement réel de l'entreprise, ou si l'on veut, les règles du jeu implicites des rapports entre les hommes, les groupes et les catégories". En particulier, la perte de la maîtrise de ressources informationnelles dont disposaient les utilisateurs peut susciter une opposition résolue d'agents économiques qui pouvaient valoriser certaines asymétries informationnelles (Eric Brousseau, 1993, p. 226). Les logiciels peuvent souvent permettre l'instauration de nouvelles méthodes de contrôle, beaucoup plus efficaces, de l'activité réalisée (Robert Michon et Lin Gingras, 1988). Ils nécessitent l'accomplissement de nouvelles tâches,

¹³⁴ Cf. l'exemple des ouvriers d'entretien qui avaient fait disparaître les notices des machines pour préserver leur pouvoir, analysé par Michel Crozier (1963).

imposant de nouvelles contraintes dans l'activité (exactitude, vigilance, réponse en temps réel), qui ne sont pas toujours reconnues au niveau de pénibilité qu'elles impliquent (*stress*). En même temps, ils peuvent représenter le risque, en codifiant dans les logiciels des connaissances tacites détenues par les opérateurs, de permettre de ne plus avoir besoin de certains d'entre eux ou de rendre obsolètes certaines de leurs compétences. Analysant de façon plus générale, la capitalisation des connaissances, Anne Mayère se demande "s'il est raisonnable et réaliste" de supposer que les acteurs et les collectifs dans l'entreprise vont contribuer à un "modèle organisationnel qui vise à les instrumentaliser toujours plus et qui, en leur demandant de formaliser leurs savoirs, se donne des moyens supplémentaires de se passer de leurs services" (1997, p. 206). Cet "effort de rationalisation et de formalisation des savoirs mobilisés dans l'entreprise, (...) projet taylorien poussé à son extrême, en étendant le concept de "tâches" à toutes les activités de l'entreprise" (idem, p. 205) a été analysé précisément par Yvette Lucas (1989) dans le cas de l'aéronautique, comme représentant un "vol de savoir" ; en effet, la codification des connaissances dans un logiciel entraîne un transfert de "propriété" des connaissances codifiées vers ceux qui détiennent les droits de propriété sur le logiciel.

Enfin, même "dans l'hypothèse particulièrement favorable de consensus sur les bénéfices à retirer du changement", Olivier Favereau montre, en s'appuyant sur différents exemples et analyses, la nécessité, pour enclencher des dynamiques de transformation, "d'établir une confiance réciproque cumulative entre dirigeants et exécutants" ; cette confiance est difficile à construire car "les caractéristiques mêmes du pouvoir, qui se révèlent si efficaces pour produire la conformité, sont contradictoires avec les mécanismes de production de la confiance" (1998, p. 225-228).

Il apparaît donc qu'il est particulièrement difficile pour des raisons simultanément communicationnelles, cognitives et sociales, de construire l'intercompréhension entre les concepteurs et les utilisateurs, dont dépendra pourtant ultérieurement la qualité des "communications" entre les logiciels et leurs utilisateurs.

2 - Les problèmes de "communication" entre les logiciels et leurs utilisateurs

L'utilisation de la plupart des logiciels implique l'existence de "dialogues" avec les utilisateurs. Un programme d'ordinateur est "un texte écrit, destiné à résoudre "interactivement" (l'ordinateur répond et pose des questions), un problème ou un ensemble de problèmes donnés. Pour un utilisateur, dialoguer avec un programme, c'est comme lire un livre qui répondrait aux questions qu'on lui pose" (Philippe Breton, 1990, p. 58). Derrière, ce

"dialogue" existe en réalité une forme de communication avec les auteurs du logiciel, qui ont "leur manière propre de résoudre le problème en question" (idem) avec leur style, leurs qualités et leurs défauts. Toute la difficulté est que cette communication n'est pas directe mais médiée par le logiciel. Dans l'hypothèse la plus favorable, il a pu exister une communication directe avec une partie des utilisateurs du logiciel lors de sa phase de conception. Lors de l'utilisation du logiciel les dialogues de l'utilisateur ne s'effectuent plus qu'avec le logiciel, si l'on excepte les échanges avec les concepteurs pour des améliorations ultérieures. A la différence d'un dialogue direct entre des êtres humains où des incompréhensions peuvent être surmontées par la reformulation des questions, l'introduction de nouvelles questions élaborées au cours du dialogue pour préciser les réponses, le dialogue avec le logiciel ne peut reposer que sur les questions et les réponses qui ont été enregistrées par les concepteurs *ex-ante* lors du développement du logiciel. Bernard Conein (1997) souligne l'asymétrie entre une interaction objet/humain, caractérisée par la rigidité et la fixité des instructions, et une interaction humain/humain où les divergences d'interprétation peuvent être surmontées au cours de la conversation. La perfection du dialogue avec le logiciel supposerait que les questions et les réponses soient complètement dépourvues d'ambiguïté pour les utilisateurs, ce qui est évidemment impossible. Intervient notamment de façon importante le contexte concret dans lequel est plongé l'utilisateur¹³⁵.

Philippe Breton compare la communication entre concepteurs et utilisateurs d'un logiciel à la communication qui s'établit entre l'auteur et le lecteur d'un livre (1990, p. 66). En suivant cette comparaison, il est frappant de remarquer comment cette citation de Socrate dans le Phèdre de Platon à propos des écrits semble s'appliquer parfaitement aux logiciels : "on croirait que de la pensée anime ce qu'ils disent ; mais qu'on leur adresse la parole avec l'intention de s'éclairer sur un de leurs dires, c'est une chose unique qu'ils se contentent de signifier, la même toujours ! Autre chose : quand une fois pour toutes il a été écrit, chaque discours s'en va rouler de droite et de gauche, indifféremment auprès de ceux qui s'y connaissent et, pareillement, auprès de ceux dont ce n'est point l'affaire, il ne sait pas quels sont ceux à qui justement il doit ou non s'adresser. Que d'autre part il s'élève à son sujet des voix discordantes et qu'il soit injustement dédaigné, il a toujours besoin de l'assistance de son

¹³⁵ Le rôle du contexte est décisif dans le cas de l'utilisation de logiciels "systèmes experts" qui ne peuvent être appliqués qu'à des domaines étroitement limités sans dénaturer la logique de l'expertise (Robin Cowan, Dominique Foray, 1998, p. 315).

père : à lui seul, en effet, il n'est capable, ni de se défendre, ni de s'assister lui-même" (cité par Harry M. Collins, 1992, p. 27).

Toutefois, les conséquences des imperfections de la communication entre auteur et "lecteur" sont en général plus importantes dans le cas des logiciels : dans le cas d'un livre, une mauvaise compréhension de ce qu'a voulu dire l'auteur ne peut avoir éventuellement que des conséquences indirectes, quand la lecture du livre influence les actions des lecteurs. Par contre, le logiciel étant un texte directement "actif", il comprend en quelque sorte deux types de "lecteurs" : les utilisateurs qui entrent des données et choisissent les traitements à effectuer en fonction de leur compréhension du programme, et le matériel informatique qui va exécuter automatiquement les parties du programme correspondant aux choix des utilisateurs.

C'est à ce stade de l'utilisation du logiciel que vont se manifester les difficultés d'intercompréhension concepteurs/utilisateurs analysées précédemment. "Un programme au début surprend toujours son utilisateur par son imprévisibilité apparente : les réponses de la machine sont parfois surprenantes quand on ne connaît pas le style du programme en question" (Philippe Breton, 1990, p. 69). Un exemple bien connu est le désarroi de l'utilisateur néophyte de Windows qui, pour arrêter correctement l'ordinateur, doit cliquer sur le bouton ... "Démarrer". Ce qui est considéré comme absurde par l'utilisateur apparaissait en fait logique pour les concepteurs de Windows, pour lesquels arrêter un ordinateur consiste à démarrer l'exécution d'un programme spécifique¹³⁶.

Certes ces problèmes ne sont pas spécifiques aux logiciels et existent pour tout artefact dont les possibilités d'une utilisation satisfaisante dépendent de la qualité de l'intercompréhension entre concepteurs et utilisateurs. Mais ce problème prend une importance spécifique dans le cas des logiciels, en raison du nombre très élevé des choix proposés par l'artefact à l'utilisateur. La plupart des objets n'ont qu'un nombre de fonctionnalités limitées, ce qui restreint l'ampleur des communications avec l'utilisateur. Plus précisément, la variété des utilisations de multiples objets a été étendue en les rendant

¹³⁶ De même, la fonction "aperçu avant impression" qui permet de visualiser à l'écran un document tel qu'il sera imprimé, se trouve curieusement dans le menu "Fichier" et non dans le menu "Affichage". Ce fait déconcertant pour l'utilisateur, qui veut choisir un "affichage" différent de son travail, s'explique par l'historique de cette fonction. La fonction "aperçu avant impression", qui se trouve maintenant sur la plupart des logiciels, a été à l'origine conçue par un programmeur qui mettait au point un module d'impression des fichiers, et avait pour seul objectif de s'épargner les fréquents déplacements qu'il devait effectuer jusqu'à une imprimante distante pour juger de la qualité de son module.

programmables, c'est à dire en intégrant des logiciels (le plus souvent gravés dans des composants) dans ces objets. Dans le cas d'objets "grand public", cette programmabilité a occasionné des difficultés pour l'utilisateur, dont le meilleur exemple est le magnétoscope, ce qui peut conduire à limiter la surabondance des fonctions. Par contre, dans le cas de l'ordinateur, machine universelle dont la multiplicité des usages ne cesse de s'étendre, la variété des communications va continuer à se développer, d'autant plus que les interfaces entre l'homme et l'ordinateur se diversifient de plus en plus. En même temps, la richesse de ces nouveaux canaux de communication (par exemple la parole) contribue à prêter à un système informatique des formes d'intelligence dont il est dépourvu, ce qui ne peut qu'augmenter le sentiment d'insatisfaction de l'utilisateur devant des comportements du système différents de ceux qu'il escompte¹³⁷.

Conclusion : l'économie du logiciel en crise permanente ?

En 1968, au cours d'une conférence sur le thème du génie logiciel organisée par l'Otan, des professionnels du logiciel créèrent l'expression "crise du logiciel", qui selon la plupart des études perdurerait depuis cette époque (cf. par exemple, OCDE, 1991 A, p. 41). Il est vrai que de nombreux exemples que nous avons mentionnés lors de l'étude de la productivité, de la fiabilité et de la qualité semblent confirmer la réalité de cette crise. Un aspect significatif est la proportion des projets qui doivent être abandonnés en cours de développement. Capers Jones estime que 24 % des projets sont arrêtés sans avoir été achevés (1998, p. 21)¹³⁸. Cette proportion est encore plus importante pour les grands projets : selon la même source, le taux d'annulation dépasse 50 % pour les systèmes de taille importante comportant plus de 10000 points de fonction. Frederick P. Brooks compare les développements des gros programmes

¹³⁷ Harry M. Collins souligne le danger des interprétations anthropomorphiques des systèmes informatiques, qui mettent en place des attentes de la part des usagers, qui seront forcément déçus (1992, p. 108). Certains auteurs estiment même qu'il est négatif que les ordinateurs acquièrent des compétences dans le domaine de la conversation naturelle, car cela aboutirait à les faire passer pour doués d'une expertise et d'une infaillibilité dont ils ne disposent pas en réalité.

¹³⁸ Toran Demarco, et Timothy Lister titre le chapitre I de leur livre "quelque part, en ce moment un projet est en train d'échouer" (1991, p. 12), Lauren Ruth Wiener affirme connaître "des programmeurs qui ont vingt ans de métier derrière eux et qui n'ont jamais contribué à un produit commercialisé" (1994, p. 110).

aux énormes animaux préhistoriques qui s'engluaient dans les lacs de bitume naturel : parmi ces programmes, "peu ont atteint leurs buts et respecté leurs calendriers et leurs budgets (...), la difficulté ne [semblant] pas provenir d'un point particulier (...) mais de l'accumulation de facteurs qui interagissent simultanément et ralentissent les mouvements de plus en plus" (1996, p. 4). De nombreux exemples peuvent être cités à titre d'illustration : le nouveau système de contrôle aérien aux Etats-Unis dont les travaux ont été suspendus après quelques milliards de dollars de dépenses (Jacques Printz, 1998, p. 40) ; le système financier développé pour la Bank of America d'un coût de 20 millions de dollars sur quatre ans et qui ne fonctionna que très épisodiquement pendant moins d'un an (Lauren Ruth Wiener, 1994, p. 110) ; un système de manutention automatique de bagages qui a retardé de plus d'une année l'ouverture du nouvel aéroport de Denver, système qui ne fonctionne toujours pas correctement, une partie de la manutention des bagages devant être effectuée manuellement (Maarten Boasson, 1998, p. 2).

Une deuxième manifestation (et conséquence) de cette crise est l'importance prise par la maintenance. Les estimations sont variables selon les auteurs mais toujours très élevées : une enquête menée en 1986 par *EDP Software Maintenance* auprès de 55 entreprises concluait que 53 % du budget total du logiciel est affecté à la maintenance (Patrick Jaulent, 1992, p. 17) ; Jean-Marc Geib estime que la maintenance représentait 65 % des coûts de développement des logiciels (1989, p.5), évaluation proche de celle de Michael A. Cusumano qui est de 67 % (1991, p. 65) ; Frédéric Georges Roux considérait que 75 % de l'activité des informaticiens, en 1988, était consacrée à la maintenance au détriment du développement de nouvelles applications (1991, p. 39). Comme la maintenance croît avec le stock d'applications utilisées, et que ce stock augmente en raison de la différence positive entre le l'introduction de nouveaux programmes et le retrait des anciens programmes, le Gartner Group estimait que les dépenses de maintenance pourraient atteindre dans les années à venir jusqu'à 95 % des budgets de programmation (Jean-Marie Desaintquentin, Bernard Sauter, 1991, p. 60). Il est vrai que la maintenance peut recouvrir dans le domaine des logiciels des activités plus variées, que celles habituellement désignées par ce terme. Lauren Ruth Wiener estime même que le terme de maintenance est utilisé d'une "manière totalement abusive pour désigner ce qui arrive à un logiciel une fois que le client l'a acheté. A la différence des voitures et des autres systèmes physiques, un logiciel ne s'use pas quand l'on s'en sert. Il n'a pas à surmonter frottement ou inertie. Il n'a besoin ni de graissage, ni de nettoyage. (...). Ce dont le logiciel a besoin une fois commercialisé, ce n'est pas de la maintenance, mais des réparations" (1994, p.

132). Toutefois la maintenance corrective (la correction des défauts résiduels d'un logiciel après sa mise en service) ne représente qu'une partie de la maintenance. La "maintenance" remplit également d'autres fonctions : amélioration des performances (maintenance perfective), adaptation à des modifications d'environnement comme le changement du matériel ou du logiciel système (maintenance adaptative), et surtout l'ajout de nouvelles fonctionnalités représentant des améliorations pour répondre aux besoins des utilisateurs (maintenance évolutive)¹³⁹. L'importance prise par la maintenance s'explique également par une productivité plus faible que dans le développement de nouveaux logiciels, en raison de la nécessité d'intégrer des changements à un cadre déjà existant (Capers Jones, 1989, p. 57) et, en général, les moins grandes qualifications et expériences des personnes affectées à la maintenance des logiciels, opération considérée comme peu valorisante. En conséquence la maintenance des logiciels fait croître leur entropie (Frederick P. Brooks, 1996, p. 104), rendant encore plus difficiles les opérations de maintenance ultérieures. Au bout d'un certain temps, il se produit un phénomène de vieillissement du logiciel, rendant nécessaire des "soins gériatriques" (Capers Jones, 1998, p. 26). La durée de vie des applications beaucoup plus longues que prévue, dont certaines encore en service ont été écrites dans les années soixante-dix (Luc Rubiello, 1997, p. 134), peut rendre nécessaire de reprendre complètement la conception du logiciel pour le restructurer et l'adapter (réingénierie), ce que certains auteurs considèrent comme faisant également partie de la maintenance.

La crise du logiciel semble correspondre à un cercle vicieux, dont il est difficile de sortir. Les qualités insuffisantes (fiabilité, inadaptation) des logiciels génèrent une maintenance importante, qui influe négativement sur la productivité globale du secteur. De ce fait, il existe un nombre important de développements en attente¹⁴⁰ auquel il faut ajouter, pour les grands projets, la longueur du cycle de développement, dont nous avons vu qu'elle ne pouvait être réduite en augmentant les effectifs sur un projet ("loi de Brooks")¹⁴¹. Dès lors, la

¹³⁹ Sur 15 000 programmeurs en 1983 à IBM, 10 000 étaient occupés à ajouter des fonctionnalités aux programmes existants (Capers Jones, 1989, p. 55).

¹⁴⁰ Jean-Marie Desaintquentin et Bernard Sauter soulignent que l'arriéré de demandes insatisfaites (ou *backlog*) entre les utilisateurs et le service informatique a pris aujourd'hui dans certaines entreprises "des dimensions réellement colossales" (1991, p. 61). En moyenne, Jean Brès estime qu'il existe deux ou trois ans de développements en attente (1994, p.5).

¹⁴¹ Les systèmes informatisés de réservations aériennes, qui ont mis 30 ans pour être développés, constituent un cas extrême (Godefroy Dang Nguyen, 1995, p. 346).

pression est forte de produire le plus rapidement possible un programme utilisable avec un minimum de travail préalable à la programmation (analyse des besoins, spécification, planification et conception), ce qui a évidemment des conséquences en termes de fiabilité et de qualité des logiciels produits. Elle a également pour conséquence de freiner l'introduction de méthodes susceptibles d'améliorer la productivité à moyen terme, mais dont le temps d'apprentissage et d'assimilation risque de dégrader les performances à court terme.

Pourtant, nous avons également mis en évidence l'importance des améliorations existantes, améliorations partiellement masquées par la progression concomitante des besoins en logiciels et des exigences des utilisateurs. Certains auteurs estiment que les questions critiques de l'économie du logiciel s'expliquent avant tout par la jeunesse de cette activité et que les progrès scientifiques suffiront pour résoudre la "crise du logiciel" de façon identique à ce qui s'est produit dans d'autres domaines. Par exemple, Capers Jones conclut son ouvrage par le constat que "programmation et logiciel ont encore moins de quarante ans d'âge. Le rythme du progrès qui nous entraîne de cette forme d'art intuitif, pratiqué isolément, vers une discipline d'ingénierie est en fait aussi rapide que pour n'importe quelle autre science. Un panorama du génie chimique d'il y a cinquante ans, de la géologie d'il y a cent ans et de la physique d'avant 1910 feraient ressortir des problèmes analogues à ceux qui sont aujourd'hui les nôtres" (1989, p. 296). Il nous semble au contraire qu'indépendamment des progrès bien réels existants, la production des logiciels présente des spécificités qui rend particulièrement difficile la réalisation de logiciels qui seraient conjointement très fiables, bien adaptés aux besoins des utilisateurs et produits avec de hauts niveaux de productivité. En particulier la résolution d'une des questions critiques de l'économie du logiciel rend souvent impossible des progrès identiques sur les autres dimensions. Il en résulte que dans l'économie du logiciel se développe simultanément des processus différents de rationalisation de l'activité, distincts selon les nécessaires compromis opérés entre les différentes caractéristiques de l'activité, ce qui confère à l'économie du logiciel une grande diversité.

Chapitre VI

LA DIVERSIFICATION DE L'ECONOMIE DU LOGICIEL

Hormis la permanence des questions critiques précédemment analysées, le deuxième constat majeur auquel conduit l'investigation de l'économie du logiciel est celui de sa diversité croissante. Celle-ci est le résultat d'un triple phénomène. Il s'agit, tout d'abord, de la variété croissante des logiciels. Les produits réalisés sont extrêmement divers, depuis les quelques lignes de programme écrites par un utilisateur pour ses besoins propres, jusqu'aux logiciels de contrôle de la navette spatiale, en passant par des produits aussi différents que des jeux, des traitements de texte, des systèmes de gestion d'entreprise... La diversité des logiciels augmente continuellement avec l'extension des domaines d'application de l'informatique et la création permanente de nouveaux types de produits. Comme le fait remarquer Gérard Dréan, "la variété des formes de logiciel est aussi grande que celle des formes et des utilisations possibles de l'écriture" (1996 A, p. 197). De plus, un produit aux fonctionnalités identiques peut présenter de grandes différences sur le plan économique (depuis la réalisation d'une application sur mesure jusqu'à des logiciels produits à des millions d'exemplaires) et sur le plan juridique (cf. chapitre III). Le deuxième phénomène est la diversité croissante des utilisateurs, y compris pour un même type de logiciel, qui résulte de la diffusion des logiciels dans de multiples aspects de la vie économique et sociale, ce que nous avons analysé en termes de "logicialisation" de l'ensemble des activités (cf. chapitre IV).

Le troisième phénomène, qui est tout à la fois une cause et une conséquence de la diversité des produits et des utilisateurs, concerne la diversité des producteurs. Coexistent en effet, une production réalisée en interne par les entreprises utilisatrices et une production mise sur le marché, une myriade de petites entreprises et de grandes entreprises puissantes, des entreprises spécialisées dans la production des logiciels et des entreprises pour lesquelles c'est une activité parmi d'autres. Quand on étudie l'économie du logiciel selon les dimensions habituelles de l'économie industrielle (arbitrage entre faire et faire faire, concentration, intégration, diversification, spécialisation) se manifestent des phénomènes apparemment contradictoires. C'est à l'analyse de ces phénomènes, permettant d'expliquer la diversification croissante des producteurs de logiciels que sera consacré l'essentiel de ce chapitre (section II).

Auparavant, dans une première section nous reviendrons brièvement sur la diversité des produits et des utilisateurs.

Section I - La diversité des produits et des utilisateurs

A - LA DIVERSITE DES PRODUITS SELON LEURS FONCTIONS TECHNIQUES

En 1986, Capers Jones distinguait vingt types de produits différents : produit pour utilisateurs finals (tableur, traitement de texte), programme d'application par lots, programme d'application interactif, programme d'EAO ou de support de formation, programme d'application de base de données par lots, programme d'application de base de données interactif, programme d'aide à la décision, programme de support graphique, programme de support CAO/IAO, programme scientifique ou mathématique, programme système (système d'exploitation, utilitaire, etc.), compilateur ou assembleur, générateur de programmes ou d'exploitation, programme de commande de processus, programme de communications ou télécommunications, programme intégré, programme temps réel, programme de robotique, programme d'intelligence artificiel, système militaire ou de défense (1989, p. 312). Les évolutions de l'informatique depuis cette date, et notamment le développement des réseaux, nécessitent de compléter cette liste avec l'apparition de nouveaux types de produits, qui en peu de temps ont pris une grande importance (cf. chapitre III) : logiciels de navigation sur les réseaux, moteurs de recherche, logiciels de sécurité pour les réseaux internes (*firewalls*), logiciels de cryptographie, d'authentification, logiciels permettant de traiter des données multimédias et notamment les logiciels de compression de données, logiciels de travail en groupe (*groupware*), logiciels de circulation de documents (*workflow*), agents "intelligents" (par exemple les logiciels permettant de rechercher automatiquement le prix le plus bas pour un produit donné), logiciels d'analyse de grandes masses de données (*datawarehouse*), applications partagées entre plusieurs entreprises (*crossware*)...

Parmi les logiciels, une première distinction peut être effectuée entre les logiciels de base (ou logiciels système) et les logiciels d'application¹. Les logiciels de base sont les

¹ Certaines sources ajoutent une troisième catégorie (le *middleware*) pour désigner les logiciels qui s'insèrent entre le logiciel système et les logiciels d'applications, comme par exemple les logiciels de travail en groupe (*groupware*) qui se développent rapidement.

programmes qui font fonctionner l'ordinateur sans être spécialisés pour une tâche définie, et qui sont étroitement liés au matériel. Ils remplissent des fonctions générales telles que la gestion du matériel, la gestion des tâches, la gestion des données, la gestion des communications. Ce sont principalement les systèmes d'exploitation, les logiciels de communication, les compilateurs de langages, et de multiples "utilitaires" qui apportent au matériel des fonctionnalités plus riches et plus facilement utilisables.

Les logiciels d'application personnalisent l'ordinateur en vue d'une utilisation particulière. Parmi les logiciels d'application, une distinction est parfois effectuée entre les outils d'application et les solutions d'application. Les outils d'application sont les programmes qui permettent aux utilisateurs de retrouver, organiser, gérer et manipuler des données et bases de données. Ils concernent quatre domaines : accès aux données et extraction des données, gestion des données, manipulation des données, conception et mise au point des programmes. Les principaux types de produits sont les tableurs, les systèmes de gestion de base de données (SGBD), les programmes d'aide à la décision et d'informatique décisionnelle, et les outils de génie logiciel.

Les solutions d'application sont des programmes conçus pour apporter des solutions prêtes à l'emploi à des problèmes spécifiques, inhérents à un secteur d'activité ou à une fonction particulière. Les logiciels qui assurent des fonctions "trans-sectorielles" (paye, comptabilité, gestion des ressources humaines) sont appelés logiciels horizontaux, par opposition aux logiciels verticaux qui traitent plusieurs fonctions d'un métier ou d'un secteur donnés.

En raison de la diversification des usages des ordinateurs, les logiciels d'application représentent une part croissante des logiciels :

Tableau XXXXII
Marché européen du logiciel
(en milliards de dollars)

	1994	1999	TCAM 1994-1999
Logiciels applicatifs	21,8	33,8	9,2%
Logiciels systèmes et outils	12,2	14,5	3,5%
Total	34,0	48,3	7,3%

Source : Input, in Eurostaf, 1996 A, p. 93.

Toutefois, même si leur importance quantitative relative diminue, certains logiciels systèmes ont un caractère stratégique (Christine Babelon, 1987, p. 160). C'est notamment le cas des systèmes d'exploitation, par leur situation à la charnière entre les utilisateurs, les applications et le matériel. Les systèmes d'exploitation sont "le lieu par excellence des efforts de normalisation, et l'arène où se joue la bataille des standards" (Gérard Dréan, 1996 A, p. 231). En effet, le système d'exploitation "incorpore un ensemble de conventions qui en font l'image principale de l'informatique pour chacun de ses utilisateurs ou de ses participants" (idem), et les changements concernant les systèmes d'exploitation jouent un rôle central sur l'évolution de l'ensemble de l'informatique.

La répartition entre progiciels et logiciels développés sur mesure est différente selon le type de logiciels. Si la part relative des progiciels progresse pour toutes les catégories de logiciels, elle est beaucoup plus importante pour les logiciels systèmes que pour les logiciels d'application, et au sein des logiciels d'application elle est plus forte pour les logiciels horizontaux que pour les logiciels verticaux où l'adaptation aux spécificités de l'entreprise a un rôle plus essentiel (Robert Merges, 1996, p. 272). De ce point de vue, il est intéressant de mentionner la distinction qui est opérée parmi les logiciels entre "systèmes" et "extensions" (*add-on*). Les "systèmes", qui concernent aussi bien des logiciels de base que des applications, ont vocation à couvrir l'ensemble d'un domaine en offrant à l'utilisateur une large palette de fonctions cohérentes, ce qui nécessite de définir des standards en matière de formats

de données, de communication avec l'utilisateur, de communications entre composants du système. Les "extensions" ont pour objectif de réaliser une fonction particulière, et sont le plus souvent le résultat d'une innovation, qui sera éventuellement reprise dans une version ultérieure d'un "système". Les "extensions" doivent se conformer aux standards des "systèmes" correspondants. Elles sont moins complexes à réaliser et représentent des développements moins élevés pour mettre au point le logiciel que les "systèmes". Il n'est donc pas surprenant que l'échelle de production et la concentration soient plus importantes dans le cas des "systèmes". D'une certaine façon, on peut considérer que les applications sont des extensions du logiciel de base (Gérard Dréan, 1996 A, p. 213).

Enfin, deux autres facteurs contribuent à augmenter la diversité des produits fournis par les producteurs de logiciels. Il s'agit, d'une part, de la fourniture conjointe d'une gamme de services de plus en plus étendue. Par exemple, Oracle, premier fournisseur mondial de logiciels de gestion de bases de données, réalise près de la moitié de son chiffre d'affaires dans des services liés à la vente de ses logiciels (Eurostaf, 1996 A, p. 122). D'autre part, de plus en plus de logiciels intègrent dans leur production des activités diverses : production de "contenus" et présentation graphique dans le cas des sites Web ou des logiciels multimédias, activité de conseil aux entreprises pour la réalisation de systèmes d'informations ou de sites de commerce électronique...

B - LA DIVERSITE DES UTILISATEURS

Le deuxième axe de la diversité de l'économie du logiciel concerne la grande hétérogénéité des utilisateurs, y compris pour un logiciel d'un type donné. Les différentes catégories d'utilisateurs ne se servent pas de façon identique de l'ensemble des fonctions, n'accordent pas la même importance aux diverses caractéristiques d'un logiciel, et peuvent utiliser un même logiciel pour des activités variées : emploi du logiciel en l'état, réagencement pour acquérir un degré plus élevé de spécification, voire utilisation des éléments du logiciel pour produire de nouveaux logiciels, dans un objectif d'amélioration ou de fonctionnalités nouvelles (Marie de Besses, 1999, p. 264).

Cette diversité des utilisations des logiciels renvoie à la nature de l'ordinateur comme machine universelle, multifonctionnelle, et dont les relations avec l'utilisateur peuvent être très différentes : "secrétaire pour un cadre, patron pour une secrétaire, outil manuel pour un ouvrier ou un technicien..." (Jean-Pierre Faguer, Michel Gollac, 1997, p. 111). Pour la

majorité des utilisateurs dont les rapports avec l'informatique s'apparentent à un banal usage d'outils, on peut considérer qu'il s'agit d'une simple "consommation" de logiciels. Par contre, pour les informaticiens (au sens large de ceux qui s'intéressent à l'informatique), il s'agit d'une "pratique" soumise à des formes particulières de rationalité: interviennent non seulement les caractéristiques du produit, mais également le système de valeurs propres de l'utilisateur, lié à la nature émotionnelle et passionnelle de cette pratique. Cette diversité des utilisateurs croît avec la diffusion des ordinateurs au-delà de l'univers professionnel original : un micro-ordinateur sur deux vendu dans le monde actuellement est à destination familiale, suivant l'évolution des Etats-Unis où c'est déjà le cas depuis 1995.

Pour appréhender cette diversité, des typologies d'utilisateurs ont été construites : Gérard Dréan (1996 A, p. 200) distingue les auteurs-utilisateurs, les utilisateurs qui sont distincts de l'auteur mais connus de celui-ci (logiciel sur mesure) et les utilisateurs anonymes (progiciels). Pour les progiciels, Frederick P. Brooks répartit les utilisateurs en quatre niveaux : l'utilisateur de base qui fait fonctionner l'application telle quelle ; le métaprogrammeur qui construit des fonctions au-dessus d'une application donnée, principalement pour faire gagner du temps à l'utilisateur final ; le développeur de fonctions externes qui code des fonctions additionnelles pour une application ; le métaprogrammeur qui utilise une ou plusieurs applications en tant que composants d'un système plus vaste (1996, p. 247-248). Marie de Besses distingue trois catégories d'acteurs ayant des caractéristiques socioculturelles distinctes (1999, p. 271-273). La première catégorie est constituée des "usagers" (*end-user*) qui utilisent un logiciel pour réaliser des activités qui ne font pas partie du champ technologique de l'informatique ; pour ces usagers, le logiciel utilisé est un outil qui étend les capacités de création, d'obtention et de traitement de l'information dans le domaine qui intéresse l'utilisateur (comptabilité, information scientifique, commerciale...). Le terme "utilisateurs" est réservé à la deuxième catégorie d'acteurs, ceux qui détiennent une expertise technologique et portent intérêt au champ technologique de l'informatique pour lui-même ; cette catégorie comprend des étudiants et des autodidactes passionnés, et surtout la partie des informaticiens professionnels chargés de l'entretien et de la bonne marche des systèmes informatiques (administrateurs, ingénieurs systèmes, techniciens de maintenance...). La troisième catégorie (les *concepteurs* de logiciels), qui peut comprendre des personnes appartenant également à la catégorie des utilisateurs, englobe les informaticiens qui créent des logiciels (architectes, concepteurs, programmeurs...) et qui pour ce faire, utilisent d'autres logiciels.

L'extrême diversité des produits et des utilisateurs, est tout à la fois la cause et la conséquence de la diversité des producteurs de logiciels

Section II - La diversité des producteurs

La production des logiciels est l'œuvre d'acteurs très différents : les logiciels peuvent être réalisés par des services internes des organisations utilisatrices (administrations, défense, banques, assurances, entreprises industrielles) ou être achetés à des entreprises. Dans ce cas, la commercialisation de logiciels peut être l'activité principale de sociétés de logiciels (de taille très variable), ou être effectuée par des entreprises en tant qu'activité secondaire (constructeurs de matériels informatiques et de télécommunications, sociétés de conseil, et plus récemment producteurs de produits culturels). La diversité des producteurs peut être appréhendée selon trois dimensions classiques de l'économie industrielle : le statut des producteurs qui recouvre l'opposition entre *faire* et *faire faire* (A), la taille des firmes avec les phénomènes de concentration et de multiplication des entreprises due à l'apparition de nouveaux producteurs (B), l'activité principale des producteurs avec les oppositions intégration/désintégration et diversification/spécialisation (C). La production de logiciels se caractérise par la présence de phénomènes apparemment contradictoires, pour lesquels les avancées théoriques de l'économie industrielle fournissent toutefois des éléments d'explication.

A - EN TERMES DE STATUT DES PRODUCTEURS : L'ALTERNATIVE "FAIRE" OU "FAIRE FAIRE"

La diversité des statuts des producteurs de logiciels recouvre pour une large part l'opposition entre une production réalisée en interne par l'organisation utilisatrice et une production commercialisée. Après avoir présenté un éclairage théorique et effectué un premier repérage statistique de cette question (1), nous essaierons d'expliquer pourquoi l'activité de production des logiciels était à l'origine une activité essentiellement interne aux organisations (2), pourquoi elle a connu une externalisation progressive (3), mais également pourquoi il est peu vraisemblable qu'elle soit totalement externalisée (5), contrairement aux prévisions des adeptes d'un *facilities management* généralisé qui ne lui voient que des avantages (4).

1 - Cadre théorique et repérage statistique

Dans un article de 1937, Ronald H. Coase tente de fonder théoriquement l'existence des "méthodes alternatives de coordination de la production" que sont le marché et la firme. En effet, si l'on considère que la coordination marchande est *a priori* optimale, il en découle logiquement que "régulée par le mouvement des prix, la production pourrait avoir lieu sans organisation du tout" (Ronald H. Coase, p. 136²). S'il existe néanmoins des firmes, c'est en raison de la présence d'un "coût lié à l'utilisation du mécanisme des prix" qui correspond à "la découverte des prix adéquats" (idem, p. 139), à l'impossibilité de toujours établir des contrats explicites, et à l'existence de taxes et de barrières tarifaires sur le marché. C'est un calcul rationnel de minimisation du coût des échanges qui explique le recours à un mode de coordination administratif (entreprise). Si toutefois ce mode de coordination ne s'étend pas jusqu'à supprimer tout recours à des mécanismes marchands, c'est parce qu'il existe des coûts d'organisation pour effectuer les transactions à l'intérieur de la firme. C'est donc finalement la comparaison entre ces coûts d'organisation, qui augmentent avec la taille de l'organisation en raison "des rendements décroissants du management", et les coûts de fonctionnement du marché, qui détermine le mode de coordination choisi : "une entreprise tendra à s'agrandir jusqu'à ce que les coûts d'organisation de transactions supplémentaires en son sein deviennent égaux aux coûts de réalisation de cette même transaction par le biais d'un échange sur le marché" (idem, p. 143).

En 1975, Oliver E. Williamson va systématiser ce raisonnement en approfondissant l'analyse des coûts de fonctionnement du marché qu'il nomme "coûts de transaction". La coordination hiérarchique est plus efficiente que le marché lorsque les coûts de transaction sur le marché deviennent trop élevés. Oliver E. Williamson définit les coûts de transaction comme la somme des "coûts *ex ante* de négociation et de rédaction du contrat qui relie les deux entités" et des "coûts *ex post* d'exécution, de mise en vigueur, et de modification du contrat lorsque apparaissent des conflits" (1981, p. 1544). L'importance de ces coûts s'explique par le fait que les transactions sont réalisées dans un univers économique caractérisé par "l'incertitude et/ou l'existence d'un petit nombre d'échangistes qui occasionnent les défaillances du marché", associé à la "rationalité limitée" et au comportement "opportuniste" des agents économiques (Oliver E. Williamson, 1975, p. 7). Ces coûts sont

² Les références correspondent à la traduction française de 1987.

plus ou moins élevés selon la nature de la transaction et du produit échangé. Interviennent la fréquence des transactions, le degré et la forme d'incertitude à laquelle elles sont sujettes, et surtout le niveau d'investissement en actifs spécifiques qu'elles requièrent. La spécificité des actifs fait référence au degré selon lequel un actif peut être redéployé pour un autre usage et par un autre utilisateur sans sacrifier sa valeur productive. Plus ces actifs sont spécifiques, et plus il sera judicieux d'intégrer la production plutôt que de recourir au marché.

C'est à partir de ce cadre théorique³ que nous allons analyser les évolutions de l'économie du logiciel concernant cette question après avoir tenté un premier repérage statistique. Il est en général difficile d'évaluer l'importance respective de la production commercialisée et de la production réalisée en "interne", dans la mesure où on dispose rarement d'indications statistiques sur cette dernière, car elle ne fait pas l'objet de transactions marchandes. En 1991, l'OCDE considérait que "le logiciel est plus une activité diffuse qu'une branche d'activité distincte" et évaluait la part réalisée par les utilisateurs à 80 %, tout en soulignant que cette part était en baisse constante (OCDE, 1991 A, p. 19 et 35). Aux Etats-Unis, David C. Mowery (1996) estimait que, au milieu des années quatre-vingt, les organisations utilisatrices réalisaient 37 % de la production des logiciels. Le reste se répartissait entre les SSII (37 % également), les constructeurs informatiques (13 %) et les "éditeurs" de logiciels ou *Independent Software Vendor* (13 %).

³ Il ne représente qu'une partie limitée des travaux de Oliver Williamson, qui ont par ailleurs fait l'objet de nombreuses critiques (cf. Hubert Gabrié et Jean-Louis Jacquier, 1994, Benjamin Coriat et Olivier Weinstein, 1995, Nicolas Postel, 2000).

On peut tenter d'estimer indirectement l'importance de la production réalisée en interne, et surtout son évolution, en examinant où travaillent les informaticiens :

Tableau XXXXIII
Répartition des informaticiens salariés du privé
selon le secteur d'activité

Activités économiques	Effectifs		Répartition (en %)	
	1993	1997	1993	1997
Activités de services informatiques	76 100	94 800	27,7%	33,7%
Fabrication de machines de bureau et de matériel informatique	14 700	10 800	5,3%	3,9%
Autres industries et construction	65 800	61 500	24,0%	21,9%
Commerce	20 800	28 900	7,6%	10,3%
Intermédiation financière	18 700	18 000	6,8%	6,4%
Autres services	78 700	67 000	28,6%	23,8%
Ensemble des secteurs	274 800	281 000	100,0%	100,0%

Source : Insee – Enquête Emploi

La part des logiciels qui n'est pas réalisée par les entreprises utilisatrices est certainement supérieure au pourcentage des informaticiens qui travaillent dans le secteur des technologies de l'information. En effet, il est vraisemblable qu'une part plus importante des informaticiens a une activité autre que le développement de logiciels dans les entreprises extérieures à l'informatique, et que la productivité pour la production des logiciels est supérieure dans le secteur informatique, car une partie de cette production consiste à développer des progiciels et non des logiciels sur mesure. Quoiqu'il en soit, le constat le plus intéressant est la forte augmentation de la proportion des informaticiens travaillant dans les "activités de services informatiques", confirmant la tendance à l'externalisation des fonctions informatiques.

2 - Une production essentiellement interne à l'origine

Aux débuts de l'informatique, les logiciels sont le plus souvent développés par les services informatiques internes des organisations utilisatrices. C'est le cas de la plupart des logiciels applicatifs. Pour les logiciels systèmes et outils, les développements s'effectuent fréquemment en collaboration avec les constructeurs de matériels informatiques, notamment par la mise en place de groupes d'utilisateurs (*share* aux Etats-Unis), mais cette activité ne fait pas l'objet d'une facturation séparée de la fourniture de matériels de la part des constructeurs. "Dans l'histoire de l'informatique, "hardware" et "software" ont été initialement liés sur un seul marché, celui du hardware" (Christian Genthon, Denis Phan, 1999, p. 170). On peut noter que cette première phase d'informatisation se caractérise par une réinternalisation d'une partie des activités (par exemple la comptabilité, la paye des salariés) confiées à des prestataires extérieurs, le *Service Bureau*. Si à l'origine de l'informatique, les sociétés spécialisées dans le *Service Bureau*, qui sont parmi les premières à s'équiper en moyens informatiques, connaissent un certain essor grâce au partage de leurs matériels informatiques et de leurs compétences entre leurs clients, l'équipement en ordinateurs d'une proportion rapidement croissante d'entreprises entraînera la disparition des entreprises de *Service Bureau* ou leur reconversion en sociétés de services informatiques.

3 - Une externalisation progressive

L'externalisation de la production des logiciels va s'effectuer en plusieurs étapes, qui se chevauchent partiellement. De façon schématique, on peut considérer que le moment fort de chacune de ces étapes correspond à une décennie.

La première étape, située principalement dans les années soixante-dix, s'explique par l'élargissement des tâches effectuées avec l'outil informatique, permis par la baisse des prix du matériel. L'existence d'un besoin de plus en plus important de logiciels conduit un grand nombre d'entreprises à recourir à des ressources extérieures pour combler un vide temporaire (Serge Bouchy, 1994, p. 295). Cette situation constitue une opportunité pour la création de nombreuses SSII ou la reconversion dans cette activité des entreprises de *Service Bureau* et des entreprises qui vendaient du temps machine. L'organisation utilisatrice reste cependant le plus souvent maîtresse du développement des logiciels, qui s'effectue fréquemment dans ses locaux et sur son matériel, en intégrant simplement des compétences extérieures, sous forme de travail en régie. Pour certains auteurs, l'activité de ces SSII s'apparente à du travail temporaire ou de l'intérim de travailleurs très qualifiés. Il s'agit en effet, moins d'un

changement dans la nature technico-économique des logiciels produits et de leur processus de production, que d'une tentative de maîtriser les coûts de cette activité en introduisant de la flexibilité et de l'adaptabilité dans des contextes très instables. Cette première forme d'externalisation peut se combiner avec la délocalisation d'une partie de l'activité, notamment le codage, vers des pays qui ont une longue tradition d'excellence en logique et mathématique et où les politiques gouvernementales de développement des études informatiques conduisent à l'existence d'une main d'œuvre hautement qualifiée mais d'un coût beaucoup plus faible (l'Inde⁴ et plus récemment les pays de l'Est)⁵. Ce mouvement de délocalisation, déjà ancien dans la production de logiciels, est l'objet actuellement de tendances contradictoires : la nécessité de développer des relations étroites avec les utilisateurs à tous les stades du processus de production et la moindre importance prise par le codage (extension de l'automatisation et de la réutilisation) freinent ce processus ; par contre, les possibilités qu'apporte le développement des réseaux informatiques pour la circulation des produits intermédiaires de l'activité⁶, peuvent constituer de nouvelles opportunités pour une coopération dans la production de personnes dispersées géographiquement.

La deuxième étape qui recouvre principalement les années quatre-vingt, voit une extension de l'externalisation, avec un rôle plus important joué par les SSII au détriment des organisations utilisatrices. Les SSII se dotent de compétences particulières dans un ou plusieurs domaines, et ce sont ces compétences génériques qu'elles adaptent aux besoins précis de l'organisation utilisatrice. Une partie croissante des prestations réalisées devient moins spécifique aux besoins singuliers de l'entreprise cliente. Dans de nombreux cas, ces

⁴ Cette délocalisation "soft" (ou *off shore programming*) a concerné particulièrement Bangalore, dans l'Etat de Mysore en Inde, considérée comme la Silicon Valley indienne. C'est à Bangalore que Oracle, Novell et Siemens ont établi de grandes filiales de génie logiciel, et que Wipro Infotech a remodelé le système interne de General Electric en 1996 (OCDE, 1998 A, p. 48). Un informaticien indien (à compétence égale) est considéré comme coûtant 5 à 8 fois moins cher que son homologue américain ou européen (Eurostaf, 1996 C, p. 23) ; Jacques Printz estime même que le rapport est de 10 à 50 (1998, p. 329). Daniel Weissberg (1992, p. 350) parle des "cerveaux-tournevis" de la société informationnelle pour désigner cette forme de délocalisation d'une activité hautement qualifiée. Toutefois le développement impressionnant de l'industrie indienne du logiciel fait que ce sont maintenant parfois des sociétés indiennes de logiciels qui achètent des entreprises de logiciel occidentales : entre mai et septembre 2000, trois sociétés indiennes ont racheté leurs concurrents américains (Transfert, 21 septembre 2000).

⁵ Paradoxalement, l'obsolescence des machines sur lesquelles les développeurs ont l'habitude de travailler peut constituer un atout incitatif, car elle oblige les informaticiens à optimiser la gestion de leurs codes (Daniel Weissberg, 1992, p.99).

⁶ Par exemple, en Inde l'existence des "Software Technology Parks" zones industrielles aménagées par l'Etat disposant de lignes de communication à très haut débit.

SSII sont l'émanation de services informatiques de grandes entreprises, qui ont filialisé cette activité pour rentabiliser l'expertise acquise.

La troisième étape, qui est particulièrement marquante dans les années quatre-vingt-dix, correspond à la place croissante prise par les progiciels, au détriment des logiciels sur mesure. Par définition, un progiciel étant une application dotée de caractéristiques suffisamment générales pour être employée par un grand nombre d'utilisateurs, son acquisition suppose de recourir à un fournisseur extérieur. C'est l'aboutissement de la transformation d'un produit spécifique à l'origine, en un produit *générique*. La forte croissance des progiciels est liée à l'avènement de la mini informatique, des stations de travail et surtout des micro-ordinateurs. Les progiciels ont permis d'étendre le champ de l'informatique à de nouveaux usages (CAO pour les stations de travail, traitement de texte, tableur pour les micro-ordinateurs) et à de nouveaux utilisateurs (PME, simples particuliers) qui ne possédaient pas de compétences informatiques mais qui pouvaient néanmoins mettre en œuvre eux-mêmes la solution correspondant à leurs besoins. Les phénomènes de standardisation dans l'ensemble de l'informatique⁷, qui élargissent les marchés potentiels et, de façon liée, le coût beaucoup plus faible des solutions progicielles⁸ ainsi que leur disponibilité quasi-immédiate, étendent progressivement le champ des progiciels dans l'ensemble de l'informatique en se substituant aux logiciels sur mesure. Ce processus s'effectuera d'abord pour les logiciels outils et systèmes, puis pour des applications de plus en plus diverses.

En extrapolant la poursuite inéluctable de cette externalisation, certains auteurs prédisent la fin de toute activité interne des organisations utilisatrices concernant les logiciels, notamment par la généralisation du *facilities management*⁹.

⁷ Par exemple, la montée en puissance de systèmes ouverts comme Unix au détriment des systèmes "propriétaires".

⁸ Le coût d'une application spécifique est estimé 5 à 10 fois plus élevé que celui d'un progiciel (Eurostaf, 1995 B, p. 45). Il faut également prendre en compte le coût plus élevé de la maintenance : le rapport coût de maintenance / coût de développement est estimé à 1,87 contre 0,5 pour un progiciel (idem).

⁹ Par exemple, Abdelaziz Mouline, estime que "le *facilities management* représente, avec les réseaux et le *downsizing* (recul des grands systèmes au profit des puissants micro-ordinateurs et de l'informatique distribuée), l'une des trois tendances majeures qui déterminent l'évolution de l'industrie mondiale des logiciels et services informatiques" (1996, p. 68).

4 - Les avantages escomptés du *facilities management*

Le *facilities management* (ou infogérance) consiste pour le prestataire à reprendre la totalité de l'informatique du client pour une durée indéterminée. L'entreprise cliente transfère ses préoccupations informatiques vers un fournisseur, qui les prend en charge dans des conditions de service, de délais, et de prix connues à l'avance, ce qui sous-tend un engagement de résultat. C'est le prestataire et non l'utilisateur qui possède les moyens informatiques et en assume la responsabilité permanente d'exploitation pour le compte de ses clients. La société de services rachète au client les matériels installés¹⁰, embauche son personnel et s'engage à fournir un niveau de service spécifié par contrat moyennant une redevance périodique forfaitaire. Ce type de prestation, introduit au départ par EDS, est souvent présenté comme une voie d'évolution naturelle, avantageuse pour le client et inéluctable à terme.

En effet, le *facilities management* devrait permettre pour l'entreprise qui y a recours "l'optimisation de l'effort informatique" (Serge Bouchy, 1994, p. 298). Tout d'abord, il s'inscrirait dans un mouvement plus général de recentrage des entreprises sur leur "cœur de métier" (*core business*). Dans un contexte marqué par une certaine exacerbation concurrentielle, les entreprises définiraient de manière restrictive leur domaine d'activité stratégique, l'informatique et dans une acceptation plus large le système d'information n'étant plus considéré comme faisant partie intégrante du cœur de métier de l'entreprise, mais seulement comme un moyen permettant l'atteinte des objectifs stratégiques de cette dernière (Eurostaf, 1995 C, p. 2). L'externalisation généralisée de son informatique permettrait à l'entreprise de ne plus salarier d'informaticiens, "population atypique dans l'entreprise" (Serge Bouchy 1994 p. 298) et dont le pouvoir s'était accru fortement. De plus, le recours à un prestataire unique éviterait que les différents intervenants internes et externes se renvoient les responsabilités en cas de problème.

Le deuxième avantage attribué au *facilities management* est que dans un contexte marqué par l'augmentation des coûts globaux du système d'information et l'accélération du rythme de renouvellement technologique, il permettrait de transformer des coûts fixes en coûts variables et d'alléger les immobilisations de l'entreprise. Il accentuerait la souplesse et la réactivité du système d'information de l'entreprise (Eurostaf, 1995 C, p. 7) par rapport aux

¹⁰ Les matériels peuvent être sur site client, sur site prestataire avec machines dédiées au client, ou sur site prestataire avec machines partagées entre plusieurs clients.

changements technologiques mais aussi économiques (par exemple la nécessité d'harmoniser les systèmes d'information lors des opérations de fusion-acquisition plus fréquentes et plus internationalisées).

Le troisième avantage, qui est le plus souvent mis en avant par les prestataires, est que le *facilities management* entraînerait une réduction des coûts pour l'entreprise cliente, le prestataire d'infogérance pouvant réaliser des économies d'échelle, hors de portée d'un client, fût-il important, par le partage de ses moyens matériels et humains entre plusieurs de ses clients (Eurostaf, 1995 A, p. 24). Ceci implique toutefois que l'entreprise prestataire soit une société de service de très grande taille, disposant de compétences thématiques, sectorielles, techniques et méthodologiques très évoluées, d'une capacité de financement importante et d'une assise technologique sécurisante (Serge Bouchy, 1994 p. 298). Selon Eurostaf, seules les très grandes sociétés américaines (IBM/Axone, EDS et Computer Sciences) en raison de leur taille, de leur surface financière et de leur couverture géographique peuvent proposer ce type de prestations pour l'ensemble des secteurs d'activité, les entreprises d'une taille moindre devant se spécialiser sur des niches sectorielles : par exemple, en France, GSI pour la distribution, Sligos, SG2 et Axime pour la banque (Eurostaf, 1995 C, p. 23, et Eurostaf, 1995 A, p. 26) ou, aux Etats-Unis, Policy Management Systems et Continuum pour l'assurance, Shared Medical Systems et HBO & Company pour la santé...

Ces avantages semblent confirmés par la croissance du marché du *facilities management* notamment aux Etats-Unis et en Grande-Bretagne, où l'infogérance s'étend au secteur public, en étant même parfois effectué par des entreprises privées d'origine étrangère (EDS, IBM, Andersen, SEMA, CAP...) ¹¹. Si le *facilities management* semble moins développé en France, et concerne essentiellement des grosses PME industrielles et commerciales, et des banques de taille moyenne, cette situation est souvent analysée en termes de retard qui devrait être comblé par la forte croissance prévue à la fin des années

¹¹ Eurostaf cite l'exemple d'un contrat de 10 milliards de F sur dix ans conclu par EDS avec l'administration fiscale britannique (1995 C, p.12). De même, le ministère de l'intérieur anglais a externalisé toute son informatique de gestion à la société franco-britannique Sema Group (Abdelaziz Mouline, 1996, p. 70).

quatre-vingt-dix de l'infogérance (un taux de croissance annuel moyen anticipé de 18 % selon Input)¹².

5 - Les limites d'une externalisation généralisée et le nécessaire maintien d'une activité interne complémentaire

Cependant, il n'est pas évident que la place qu'occupe, actuellement et dans un futur prévisible, le *facilities management* soit aussi importante que ce que laisse supposer la lecture de la presse et de la littérature spécialisée. Selon Gérard Dréan, les véritables prestations d'infogérance globale, restent limitées à un petit nombre de cas spectaculaires et fortement médiatisés¹³, et n'ont pas connu le développement annoncé (1996 A, p. 282). Par contre, le *facilities management* étant un "concept" à la mode, il est fréquemment utilisé par des entreprises commercialement intéressées, pour désigner des prestations moins radicales, comme l'infogérance dite "de transition" (car de durée limitée) ou l'infogérance partielle réduite à l'exploitation du site central ou du réseau, ou encore de simples services de support à l'exploitation comme la gestion de parc ou la maintenance des applications (idem, p. 283).

C'est ainsi qu'une étude d'Eurostaf consacrée à l'infogérance, estime qu'il "y a *facilities management* lorsque le client transfère la responsabilité de la gestion au quotidien de l'optimisation de tout *ou partie* de son informatique auprès d'un prestataire extérieur" (Eurostaf, 1995 C, p. 7). Est par exemple appelée infogérance micro, l'optimisation, l'exploitation et la maintenance des parcs de micro-ordinateurs, segment particulièrement porteur (la progression annuelle moyenne prévue est de 29 % sur les cinq années à venir), en raison du besoin de rationalisation des parcs de micro-ordinateurs après plusieurs années d'acquisition d'équipements de manière anarchique (Eurostaf, 1995 A, p. 26-27)¹⁴. De plus, la plupart des prestations d'infogérance sont réalisées en régie, les prestations tarifées au forfait

¹² Selon Input, sur le plan européen, l'infogérance représentait 2,5 % du marché des services informatiques en 1993, 3,2 % en 1994 et la poursuite de cette forte croissance devait lui permettre d'atteindre 5,9 % en 1999 (Eurostaf, 1996 A, p. 25-26).

¹³ Par exemple le contrat de 16 milliards de francs sur une période de dix ans signé par IBM avec le constructeur aéronautique McDonnell Douglas pour gérer l'ensemble de son système informatique et de télécommunications.

¹⁴ Le *facilities management* micro en homogénéisant le parc micro et les logiciels utilisés devrait permettre d'abaisser le coût total d'un PC de 20 à 30 % (d'après Thoaminfor Spectral), notamment en réduisant les coûts cachés liés à la méconnaissance des logiciels et matériels, qui s'élèvent selon le Gartner Group à 50 000 F par an (Eurostaf, 1995 C, p. 6).

("forfait d'exploitation") représentant moins de 5 % de l'ensemble. Les contrats de gestion, par un prestataire extérieur, s'accompagnent encore souvent en France d'une tarification "au volume" et ne comportent pas de clause de prise de responsabilité¹⁵. Enfin, il existe peu de contrats d'infogérance pluriannuels en France (Eurostaf, 1995 C, p. 10).

Tableau XXXXIV
Structure du marché français par type d'infogérance
(en 1994)

	en millions de F.	en pourcentage
Infogérance globale	2 930	32,2 %
Infogérance d'exploitation	2 903	31,9 %
Infogérance de solutions	2 830	31,1 %
Forfait d'exploitation	437	4,8 %
Total	9 100	100,0%

Source : données P.A.C. citées in Eurostaf (1995 C, p. 11)

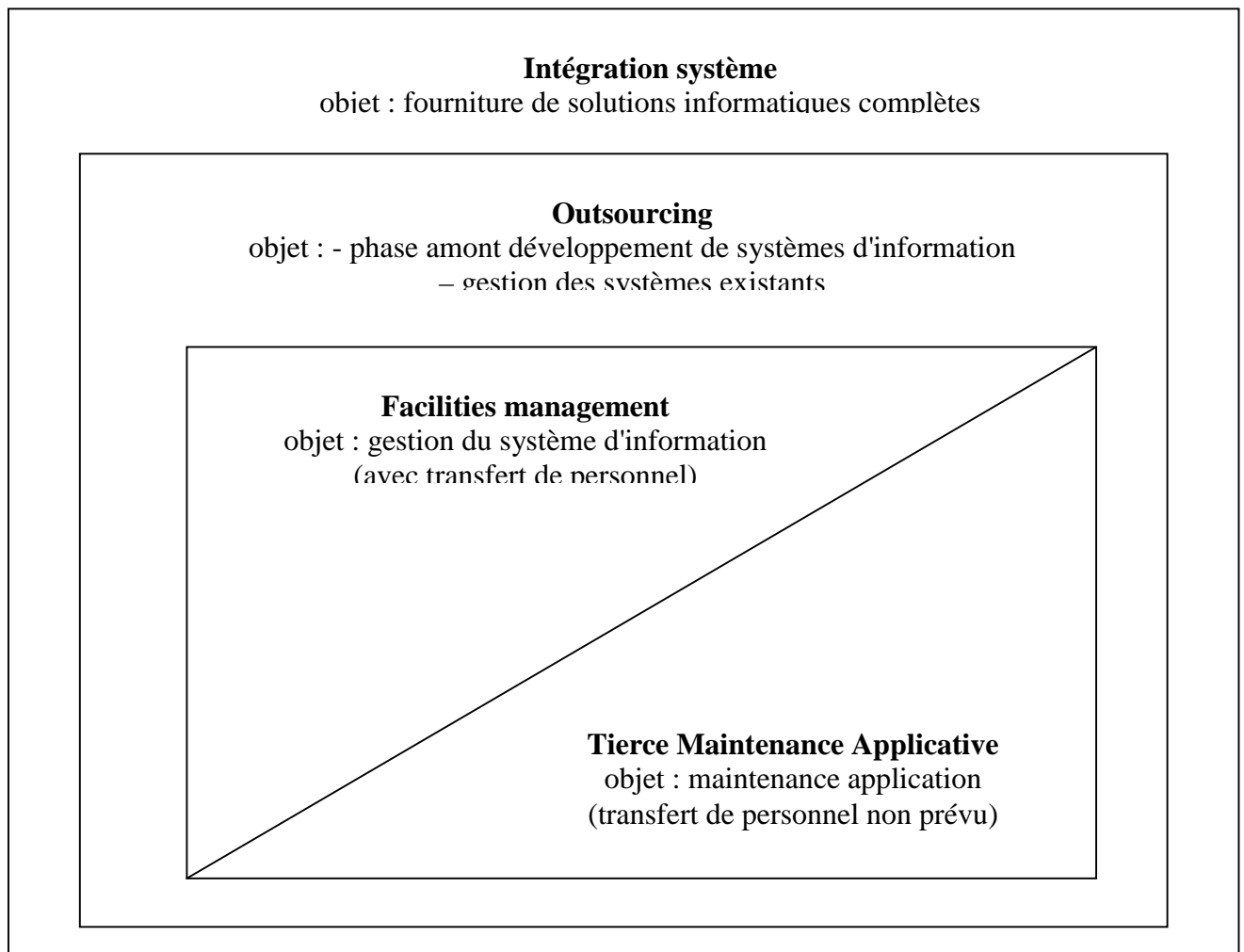
L'infogérance en régie se répartit en trois composantes de poids identique : *l'infogérance globale* réside dans la prise en charge par le prestataire d'infogérance de l'exploitation d'un ou de plusieurs centres de traitements (qui restent la propriété du client), au moyen d'apports par le prestataire des ressources humaines et méthodologiques nécessaires, notamment, à la gestion optimale et à la sécurité. *L'infogérance d'exploitation* se définit par la prise en charge totale ou partielle de l'exploitation de la fonction informatique du client, au moyen de ressources généralement fournies par le prestataire sur son site de production. *L'infogérance de solutions* est un concept hétérogène recouvrant des solutions d'infogérance non globales, dont les contours sont adaptés aux besoins spécifiques du client (Eurostaf, 1995 C, p. 11). L'infogérance de solutions est appelée infogérance verticale quand elle consiste à externaliser toutes les fonctions concernant une application (par exemple la paye) et infogérance horizontale quand elle concerne une fonction d'exploitation pour toutes les

¹⁵ "Certaines SSII françaises déclarent se positionner sur le marché du *facilities management* alors qu'elles offrent seulement des services de délégation de personnel, c'est à dire des services de délégation de personnel spécialisé sur site client sans obligation de résultats et de fourniture de moyens, et sans prise de responsabilité de tout ou partie du système d'information" (Eurostaf, 1995 C, p. 7).

applications de l'entreprise (par exemple l'exploitation et l'administration des réseaux). Il importe de noter que ce sont ces types de prestation (*facilities management* réseaux, *facilities management* micro) qui connaissent la croissance la plus rapide (idem, p. 15).

Au fur et à mesure que ce qui est désigné comme étant du *facilities management* englobe des éléments de plus en plus variés, les frontières avec d'autres prestations, dont certaines sont anciennes et en perte de vitesse comme le "traitement de données", deviennent de plus en plus floues. Le *facilities management* s'intègre dans l'*outsourcing* qui peut désigner toutes les formes de sous-traitance. La distinction entre le *facilities management* (service de substitution), et l'intégration de système (service de support), qui est une intervention temporaire d'un prestataire pour fournir une solution informatique complète, assemblant du matériel et du logiciel de différentes origines, n'est pas toujours évidente. Il en est de même avec la tierce maintenance applicative, activité en très forte croissance, qui consiste pour le prestataire à prendre en charge la maintenance d'une ou plusieurs applications avec engagement de résultats mais sans transfert de personnel. Eurostaf positionne le *facilities management* de la façon suivante (1995 C, p. 9) :

Schéma X



Comme le souligne Gérard Dréan, "conçue à l'origine comme une tentative de réintégration de prestations éclatées en une offre unique, l'infogérance suit déjà une dynamique d'éclatement et de spécialisation" (1996 A, p. 283), et "sous sa forme la plus complète (...), où le prestataire aurait à la fois la responsabilité de l'intégration du système et de la production permanente du service, l'infogérance est en réalité extrêmement peu pratiquée" (idem, p. 317). L'explication de cet état de fait réside dans la relativisation des avantages qu'une externalisation généralisée de l'informatique est censée apporter, et dans l'existence d'inconvénients non négligeables.

Une baisse des coûts peu évidente

C'est tout d'abord le cas pour la *baisse des coûts*. Eurostaf estime que la viabilité du modèle nécessite que les coûts directs du prestataire soient inférieurs de 30 à 40 % aux propres coûts de l'entreprise (Eurostaf, 1995 C, p. 22)¹⁶. Il faut notamment intégrer les coûts de migration vers le prestataire (qui représentent au moins 20 % du contrat) auxquels il faut ajouter des coûts du même ordre, en cas de reprise d'activité en fin de contrat ou d'une sortie avant terme (idem, p. 20). Or, dans la réalité, les coûts ne sont le plus souvent pas réduits à la hauteur des prévisions des fournisseurs. En effet, une réduction des coûts pourrait s'expliquer par l'existence d'économies d'échelle, du fait de la mutualisation sur plusieurs clients des moyens matériels et humains du prestataire. En ce qui concerne les moyens matériels, le développement des systèmes distribués va à l'encontre du partage des ressources machines entre plusieurs clients, qui peuvent de plus poser des problèmes de sécurité, de performances et de gestion. En ce qui concerne les ressources humaines en général, un avantage pour l'entreprise cliente peut être de reporter sur le prestataire les problèmes sociaux inhérents à la réduction des effectifs d'exploitation (Gérard Dréan, 1996 A, p. 319). C'est du reste pendant la période où les services informatiques ont connu une récession que s'est le plus développée l'infogérance. Il faut toutefois noter que cette récession n'a été que passagère, et qu'il est également possible d'imposer des réductions d'effectifs tout en maintenant l'activité informatique à l'intérieur de l'entreprise, notamment en instaurant une contractualisation interne laissant les critères financiers jouer un rôle déterminant entre les différents services de l'entreprise, constitués en centres de profit. Reste seulement l'argument du partage d'un

¹⁶ De même, Gérard Dréan chiffre à 40 % les économies que doit réaliser le prestataire pour pouvoir "faire bénéficier son client d'une réduction de 20 % de ses dépenses informatiques, comme certains n'hésitent pas à le promettre" (1996 A, p. 317).

potentiel limité de compétences critiques, qui peut justifier de recourir à un prestataire extérieur restreint à cette activité, et non une externalisation globale.

Les difficultés pour établir un contrat global

La deuxième difficulté concerne l'établissement d'un *contrat d'infogérance globale*. Il est impossible dans de tels contrats de tout spécifier, de prévoir toutes les éventualités possibles, et les contrats d'infogérance comportent fréquemment la signature d'avenants pour clarifier les inévitables zones de flou¹⁷. Ces contrats nécessairement incomplets dans des situations caractérisées par l'incertitude et l'asymétrie d'information entre les contractants, génèrent des coûts de transaction élevés : coûts d'établissement des contrats, coûts de la surveillance de leur exécution et du pilotage, coûts des garanties dont s'entoure le prestataire pour mettre le client en confiance, coûts d'opportunité subis par le client du fait de la divergence d'intérêts qui subsiste avec son prestataire, Jacques Girin ajoutant également les coûts d'évaluation du résultat (1994, p. 23). Pour de tels contrats, on atteint rapidement une situation où le bénéfice attendu d'une amélioration des contrats devient inférieur au coût de cette amélioration. On retrouve l'argumentation de Oliver E. Williamson selon lequel les *avantages de l'externalisation diminuent quand la transaction devient plus idiosyncrasique*. Même si cette transaction est peu fréquente, voire unique, son ampleur peut justifier de mettre en place des formes organisationnelles spécifiques (Benjamin Coriat, Ollivier Weinstein, 1995, p. 59). En effet, pour un contrat d'infogérance globale, l'absence d'une *confiance* importante entre les partenaires pour gérer de façon dynamique la relation client-fournisseur, risque d'enclencher des cercles vicieux de pression sur les prix, de dérive juridique sur la limitation des contrats et de déception par rapport à l'infogérance (Eurostaf, 1995 C, p. 23)¹⁸. C'est ce qui explique que quand le choix de l'externalisation est néanmoins effectué, il s'agit fréquemment d'une situation de partenariat entre entreprises prestataires et clientes, partenariat qui peut se traduire par la création d'une filiale commune¹⁹ ou par des prises de

¹⁷ Gérard Dréan estime que 60 % des contrats de *facilities management* sont renégociés dans les deux mois.

¹⁸ En conséquence Eurostaf estime que les entreprises clientes devraient éviter de signer des contrats de *facilities management* globaux, des contrats sur cinq voire sept ans, et des contrats qui prévoient le paiement des coûts de migration dès la première année (1995 C, p. 20).

¹⁹ C'est par exemple ce qu'a effectué Rhône-Poulenc en créant en 1995 une filiale commune Eurhône avec Axone (filiale d'IBM) détenue à 50 % par chacun des deux partenaires et destinée à gérer l'activité informatique du groupe Rhône-Poulenc dans le monde (Abdelaziz Mouline, 1996, p. 65), ou encore Delta Airlines avec ATT (idem, p. 70).

participations financières éventuellement croisées (Serge Bouchy, 1994, p. 300). Le développement du partenariat, que facilite le déploiement des technologies de l'information, est destiné à introduire une certaine confiance entre partenaires que ne peut remplacer la signature de contrats "en béton" (idem). Une part importante du chiffre d'affaires des plus grandes entreprises françaises d'infogérance est en fait constitué de marché captif intra-groupe (plus des deux tiers pour Axone avec IBM, près de la moitié pour Télésystèmes avec France Télécom).

Tableau XXXXV
Importance des marchés captifs pour les plus grandes sociétés d'infogérance
(France, 1993)

	C.A Facilities Management en millions de F.	Pourcentage marché captif
GSI	1 230	69,9 %
Axone	1 188	69,9 %
EDS	1 002	20,2 %
Axime	757	n.c.
Télésystèmes	625	42,4 %
Sligos	418	28,2 %

Source : données P.A.C. citées in Eurostaf (1995 C, p. 27).
(Ces six entreprises représentaient 67 % du marché total de l'infogérance en France.)

Le caractère stratégique du système d'information

La troisième limite à une externalisation généralisée de l'activité informatique réside dans le caractère de plus en plus stratégique du système d'information pour toutes les organisations. C'est évidemment le cas pour des secteurs comme la logistique, la grande distribution, la vente par correspondance, les chaînes à enseigne, où la maîtrise de l'information est inhérente à la logique du métier. Dans ces secteurs, les performances des entreprises ont toujours été étroitement liées à leur maîtrise des flux d'informations, qui représentent soit l'interface client, soit le cœur de l'organisation interne de l'entreprise (Eurostaf, 1995 C, p. 2). Les évolutions plus récentes caractérisées par l'avènement d'une "société de l'information" ou d'une "économie fondée sur les connaissances" (cf. chapitre IV)

font que c'est maintenant dans tous les secteurs, que "les technologies de l'information chez les utilisateurs se sont stratégiquement rapprochées du cœur de leurs compétences propres et que les conséquences de l'introduction de nouveaux concepts de systèmes d'information ont, chez ces utilisateurs, des répercussions importantes quant à leur organisation, voire même quant au contenu de leurs activités" (Claire Charbit et Jean-Benoît Zimmermann, 1997, p. 2)²⁰. Dans ce contexte, "se concentrer sur son métier", argument avancé pour justifier une externalisation généralisée, est "un slogan d'autant plus creux qu'une informatique intelligente est nécessairement intégrée à tous les métiers de l'entreprise" (Gérard Dréan, 1996 A, p. 318). Les exceptions sont des situations où le système informatique est peu intégré à l'entreprise, avec des systèmes stables et peu évolutifs, autrement dit une informatique ancienne et mal utilisée, qui peuvent justifier une situation d'infogérance temporaire, permettant à l'entreprise de consacrer ses ressources informatiques internes à la refonte totale du système (*facilities management* de transition). Cette externalisation doit rester provisoire en raison des aspects stratégiques de nombre des fonctions concernées²¹, des risques concernant la confidentialité de données et d'applications essentielles, de la création "d'une dépendance exclusive vis-à-vis du fournisseur" et "d'une fuite ou d'une perte potentielle de savoir-faire fonctionnel sur les domaines externalisés" (Serge Bouchy 1994 p. 298). Ceci correspond à l'argumentation de K. Pavitt (1984) selon laquelle le choix d'intégration prévaudra quand il existe un apprentissage par la pratique ou lorsque le savoir technique de base est essentiel.

En conclusion, il est vraisemblable que dans le domaine des logiciels comme dans le cadre plus général de l'activité informatique, continuera à exister une *combinaison entre une production réalisée en interne et le recours à des prestataires extérieurs à l'organisation*²². Le "faire faire" est inhérent aux logiciels, et s'impose pour des prestations limitées dans le temps et/ou correspondant à des compétences "pointues" non disponibles à l'intérieur de

²⁰ Dans le champ de l'informatique, Claire Charbit et Jean-Benoît Zimmermann (1997) considèrent que nous sommes entrés dans "l'ère du traitement de l'information" qui aurait succédé à "l'ère des ordinateurs universels" et à "l'ère de l'informatique".

²¹ Dans son étude sur l'infogérance, Eurostaf répartit les différentes fonctions du système d'information d'une entreprise selon leur "caractère externalisable". Les plus nombreuses sont considérées comme étant "*facilities management* non envisageable" (1995 C, p. 4).

²² Une enquête sur le degré d'externalisation des services dans la Communauté Européenne montre que les services informatiques sont le seul secteur où l'origine majoritaire du service (55 %) est "l'externe et l'interne combinés", par rapport à des situations "exclusivement internes" ou "exclusivement externes" (Benjamin Coriat, Dominique Taddei, 1993, p. 167).

l'organisation, prestations pour lesquelles le terme de *facilities management* semble peu adéquat. Par contre le développement de ce qui est spécifique à l'organisation et dont l'originalité peut constituer un avantage compétitif important, requiert une réalisation par les services internes de l'organisation, en s'appuyant éventuellement sur des produits et des prestations externes.

La coexistence persistante d'une activité interne et d'une activité externe complémentaires, ne signifie nullement qu'il n'existe pas des évolutions dans ce qui relève de chaque forme d'activité. Certaines prestations (gestion du parc de micro-ordinateurs, gestion des réseaux...) se banalisent et se standardisent, et peuvent être externalisées pour bénéficier d'économies d'échelles potentielles, ces tâches relativement faciles à spécifier et à évaluer pouvant être couvertes par des contrats standards, ce qui en fait des quasi-produits (Gérard Dréan, 1996 A, p. 319). Il peut en être de même, avec l'évolution des techniques de programmation, pour certains composants logiciels standardisés. Mais réciproquement, l'évolution générale de l'informatique qui relativise l'importance des aspects proprement informatiques de plus en plus banalisés au profit des compétences dans le domaine de l'application réalisée, peut conduire à internaliser des activités effectuées auparavant par des prestataires extérieurs. En effet, dans certains des domaines de plus en plus divers couverts par l'informatisation, ce qui devient essentiel c'est la qualité de l'intercompréhension entre concepteurs et utilisateurs, dont on peut penser qu'elle est supérieure pour des personnes appartenant à la même organisation et possédant partiellement une culture et un langage communs. Par exemple, dans le domaine des systèmes d'information, Claire Charbit et Jean-Benoît Zimmermann soulignent que "les choix des firmes doivent garantir le maintien de leurs spécificités et de leur culture" et qu'il "ne s'agit donc pas d'assujettir l'organisation de l'entreprise aux caractéristiques techniques du système d'information, mais plutôt de procéder à l'inverse" (1997, p. 22). "Ainsi, fréquemment aujourd'hui, les utilisateurs élaborent-ils eux-mêmes la composition des différents éléments du système d'information", "l'utilisateur final [pouvant] choisir, à son échelle individuelle, les composantes les mieux adaptées à ses spécificités d'utilisation" (idem, p. 14 et 19). De même, dans l'industrie aéronautique, Daniel Weissberg souligne la tendance au rapatriement de certaines activités informatiques et la limitation du nombre de sous-traitants, pour ne pas alourdir des procédures de spécification logicielle de plus en plus complexes (1992, p. 327). La réflexion, dans un cadre plus général, de Bertrand Collomb (de Lafarge Coppée) constatant que "nous avons sous-traité beaucoup de choses mécaniques, mais que, dans la même période, nous avons intégré beaucoup de choses

stratégiques ou culturelles"²³, nous semble bien caractériser l'avenir des relations entre intégration et marché dans le domaine des logiciels.

B - EN TERMES DE TAILLE DES ENTREPRISES : DES PHENOMENES DE CONCENTRATION PERMANENTE ET DE CREATION INCESSANTE DE NOUVELLES ENTREPRISES SUR DE NOUVEAUX SEGMENTS

La diversité des entreprises productrices de logiciels est également considérable en ce qui concerne leur taille. Depuis ses origines, l'économie du logiciel voit la coexistence d'entreprises très puissantes et d'une multitude de petites et moyennes entreprises (1). Cette situation résulte de la puissance des forces qui poussent à la concentration, tant dans le secteur des logiciels sur mesure que dans celui des progiciels (2), et de la création de nouvelles entreprises de taille réduite au départ, sur les nouveaux segments constitués par la diversification du matériel (mini-informatique, stations de travail, puis micro-informatique) et par l'extension continue des domaines et des fonctions couverts par l'informatisation (3). Comme le résume Christian Huitema, président de l'Internet Society²⁴, "on voit les gros poissons manger les petits, mais il y a toujours autant de petits poissons". On peut toutefois s'interroger sur la pérennité de cette situation où l'éclosion de nouvelles entreprises compense les disparitions et les acquisitions - la proportion des différents types d'entreprises restant assez stables, bien que leur composition ne cesse de changer - et se demander si dans le nouveau contexte informatique, la dynamique de la concentration ne va pas l'emporter sur le rythme des créations (4).

1 - Le constat

Dans le domaine des progiciels, la création de nombreuses sociétés ces vingt dernières années ne doit pas masquer la forte concentration du secteur : les ventes des huit plus grandes entreprises représentaient 33,8 % du marché mondial des progiciels en 1996, les deux plus importantes (IBM et Microsoft) réalisant à elles seules plus de 20 % des ventes (OCDE, 1998 A, p. 43). Pour les logiciels sur mesure et les services informatiques, le contraste est net entre

²³ Dans la discussion de l'article de Jacques Girin (1994, p. 31).

²⁴ Aux troisièmes Rencontres de la société française en réseau, organisées par le chapitre français de l'ISOC (Autrans, janvier 1999).

la présence de très grands groupes et l'existence de très nombreuses SSII petites ou de taille moyenne.

Pour l'Europe, la situation est la suivante :

Tableau XXXXVI
Répartition des entreprises européennes des logiciels et services informatiques

Effectifs	Nombre de sociétés	Pourcentage
>1000	55	0,3 %
500-1000	70	0,4 %
200-500	200	1,3 %
100-200	400	2,5 %
10-100	4 000	25,0 %
<10	11 300	70,5 %
Total	16 025	100,0 %

Source : Abdelaziz Mouline, 1996, p. 37.

La répartition des SSII en France est assez semblable :

Tableau XXXXVII
Répartition des SSII en France
(en 1993)

Catégories d'entreprises	Nombre d'entreprises	Pourcentage
Grands groupes (CA supérieur à 500 millions de F., effectifs supérieurs à 1000 personnes)	25	0,3%
SSII moyennes (CA compris entre 100 et 500 millions de F., effectifs compris entre 200 et 1000 personnes)	60	0,8%
SSII intermédiaires (CA inférieur à 100 millions de F., effectifs compris entre 10 et 200 personnes)	1 115	13,9%
Petites SSII (effectifs de moins de 10 personnes)	6 800	85,0%
Total	8 000	100,0%

Source : Syntec Informatique, in Eurostaf 1997 B, p. 255.

Si on élargit l'examen au secteur des Activités informatiques (NAF 72) défini par l'INSEE, on constate le nombre élevé d'entreprises (21 074 en 1997), la très forte proportion d'entreprises de moins de 30 salariés (95 %) et l'importance de la concentration mesurée par la part réalisée par les premières entreprises :

Tableau XXXXVIII
Indicateurs de concentration
(part des n premières entreprises)

	Secteur détaillé	C.A. (n=10)	Effectifs (n=10)	C.A. (n=50)	Effectifs (n=50)
721	Conseil en systèmes informatiques	21 %	23 %	43 %	43 %
722	Réalisation de logiciels	23 %	24 %	41 %	39 %
723	Traitement de données	25 %	23 %	54 %	48 %
724	Activités de banques de données	41 %	29 %	65 %	56 %
725	Entretien et réparation de machines de bureau et de matériel informatique	39 %	40 %	68 %	68 %

Source : INSEE EAE Services. *Les entreprises des services en 1997*, p. 103.

2 - Les explications de la concentration horizontale

La concentration horizontale peut résulter de la disparition de certaines entreprises²⁵ et en conséquence de l'augmentation de la part de marché des entreprises restantes, et/ou des mouvements de fusion/absorption particulièrement nombreux dans le secteur du logiciel. Par exemple, entre 1985 et 1993, 1500 SSII ont disparu en France (Eurostaf, 1995 A), et les opérations de restructurations se multiplient : Abdelaziz Mouline a recensé entre 1987 et 1995, 400 opérations de restructuration impliquant au moins deux SSII, avec une très forte prédominance de la croissance externe : rachats, fusions, prises de participation, offre publique d'achat (1996, p. 45-46).

²⁵ Ou de leur retrait du marché considéré quand il s'agit d'entreprises ayant plusieurs activités (par exemple l'abandon par IBM de la production de systèmes d'exploitation pour micro-ordinateurs après l'échec de OS/2).

Dans le secteur des progiciels composé principalement d'entreprises américaines, les disparitions de sociétés sont également nombreuses, soit par faillite (VisiCorp qui avait commercialisé le premier tableur VisiCalc, ou Wordstar éditeur du premier traitement de texte pour micro-ordinateur), soit le plus souvent par des opérations de fusion et d'absorption. Par exemple, sur une période de moins de dix ans, Borland, au départ producteur de langages de programmation pour micro-ordinateurs, a fusionné avec la société Arisa (progiciel principal Paradox), racheté la société Ashton-Tate (progiciel dBase) qui avait auparavant absorbé Multimate (traitement de texte), vendu une partie de ses activités (le tableur QuattroPro à Novell et le gestionnaire de base de données Paradox à Corel), fusionné avec Visigenic pour donner naissance à Inprise, qui vient à son tour de fusionner avec Corel.

Le facteur principal de la concentration horizontale est la possibilité de réaliser des économies d'échelle, c'est à dire d'obtenir un coût de production unitaire plus faible quand la quantité produite augmente. En fonction des économies d'échelle potentielles, il est possible de déterminer la taille optimale d'une entreprise. Quand la taille des entreprises pour atteindre la taille optimale, augmente plus rapidement que la croissance du marché, le secteur concerné se concentre. Il importe à ce niveau d'opérer une distinction entre la production de logiciels sur mesure et des services informatiques d'une part, la production de progiciels d'autre part, la structure des coûts étant très différente dans ces deux activités.

a - Dans la production de logiciels sur-mesure

Jusqu'à une période récente, les coûts variables étaient largement prédominants dans la production des logiciels sur mesure. Pourtant la concentration était plus importante que dans la production des progiciels (Robert Merges, 1996, p. 272). Une première explication est que la productivité et la qualité des logiciels produits sont meilleurs dans les grandes entreprises (Capers Jones, 1989, p. 295-296), notamment parce que celles-ci peuvent plus facilement substituer des spécialistes à des généralistes dans de multiples domaines (test, maintenance, mesure, assurance de la qualité, rédaction technique, gestion des bases de données, planification et estimation...). En 1998, Capers Jones estime que les grandes entreprises peuvent employer au moins cent catégories différentes de spécialistes du logiciel (1998, p. 33). Michael A. Cusumano (1991) a analysé minutieusement comment quatre très grandes entreprises japonaises (Hitachi, Toshiba, Nec et Fujitsu) arrivaient à produire de façon performante du logiciel sur-mesure de haute qualité dans de véritables usines à logiciel (*software factories*).

La deuxième explication est l'existence à côté des économies d'échelle "réelles", d'économies d'échelle "monétaires" reposant sur le fait qu'une firme de grande taille est en situation de force vis à vis de ses interlocuteurs, ce qui contribue à lui conférer un "pouvoir de marché" (Jean-Pierre Angelier, 1997, p. 64-65). De plus, les grandes entreprises utilisatrices préfèrent souvent traiter avec des prestataires de grande taille, surtout pour des contrats d'un montant de plus en plus élevé, notamment lié à la recherche de solutions "globales". En particulier, une part croissante des contrats a un caractère international, ce qui donne un avantage concurrentiel aux SSII internationalisées. L'internationalisation croissante des entreprises utilisatrices a conduit à un processus d'internationalisation récent mais vigoureux, des entreprises américaines vers l'Europe (EDS a racheté sept SSII européennes entre 1991 et 1994, y compris par des OPA hostiles²⁶), des entreprises européennes vers les Etats-Unis (notamment Cap Gemini Sogeti²⁷), et surtout entre les entreprises des différents pays de la Communauté Européenne dans le cadre de l'approfondissement de celle-ci (Abdelaziz Mouline, 1996, p. 39). Le passage d'une approche nationale à une approche régionale (l'Europe) et à une approche globale (le monde), qui s'est d'abord opéré pour les équipements matériels, puis pour les logiciels de base et de communications, s'étend maintenant de plus en plus à l'ensemble de l'industrie du logiciel (Serge Bouchy, 1994, p. 276).

Troisièmement, pour l'ensemble des entreprises utilisatrices, l'incertitude majeure sur la valeur des prestations réalisées et leur caractère souvent stratégique, conduit à accorder beaucoup d'importance à la réputation et à la pérennité du prestataire, ce qui avantage les grandes sociétés par rapport aux petites moins connues et réputées plus fragiles. Les facteurs d'incertitude sont également importants pour le prestataire et font peser sur la rentabilité de chaque projet des risques, qui ne peuvent être supportés que par des entreprises d'une taille

²⁶ Les OPA hostiles sont rares concernant les SSII, la plupart des rapprochements s'effectuant par "consentement mutuel". En effet, la valeur d'une société de services repose avant tout sur son capital humain et l'éventualité d'une fuite des cerveaux est en général dissuasive. La prise de contrôle par EDS de SD-Scicom, société anglaise mais qui est également fortement implantée en France, était la première OPA hostile en Europe et permettait à EDS devenir la deuxième société dans les services en Europe derrière Cap Gemini Sogeti (Abdelaziz Mouline, 1996, p. 39).

²⁷ La première SSII européenne réalisait 72 % de son chiffre d'affaires à l'étranger en 1995. Son PDG Serge Kampf précisait : "il est impossible que Cap Gemini Sogeti satisfasse sa légitime ambition de rester l'un des leaders mondiaux sans devenir transnationale, car ses plus grands clients sont transnationaux. Pour être plus précis, j'ajouterai qu'être transnational signifie adopter une stratégie mondiale et servir des clients mondiaux, tant dans leur diversité géographique que dans leur caractère global" (Interview à 01 Informatique, 9 octobre 1992).

suffisante pour absorber les échecs éventuels en les mutualisant sur l'ensemble de leur activité (Gérard Dréan, 1996 A, p.271).

Enfin, l'évolution technologique conduit à une transformation de l'activité et de la structure des coûts qui lui est associée. Les coûts variables, constitués essentiellement des coûts du personnel assumant directement les prestations, fréquemment sur le matériel de l'entreprise cliente, deviennent relativement moins importants au profit de coûts fixes : nécessité d'allouer des ressources à la recherche-développement et à la formation pour rester à un niveau suffisant d'expertise technologique, élaboration de méthodes de développements plus sophistiquées, investissement coûteux dans des ateliers et outils de génie logiciel et dans les matériels complémentaires, développement de composants logiciels réutilisables à l'intérieur de l'entreprise²⁸. Il en résulte de nouvelles sources d'économie d'échelle et une élévation de la taille critique des entreprises du secteur.

b - Dans la production de progiciels

Par contre, dans la production de progiciels, en raison de la nature particulière de cette activité, les coûts fixes ont toujours représenté une part prépondérante du coût total, génératrice de fortes économies d'échelle potentielles.

Alfred D. Chandler a mis en évidence l'écart qui pouvait exister entre les économies d'échelle potentielles, permises par la technologie utilisée, et les économies d'échelle réalisables "à un moment donné et dans un lieu donné", qui doivent intégrer les coûts des transports et des communications pour la production des biens comme pour leur distribution sur un marché. Il souligne dans l'explication du mouvement de concentration des dernières décennies du XIX^e siècle, l'importance des nouvelles opportunités créées par la révolution industrielle dans le domaine des transports et des communications (1992, p. 56). De même, la notion de "taille optimale" reflète "non seulement l'état de la technologie existante mais aussi l'anticipation de la taille des marchés (...) et l'élasticité de la demande" (idem, p. 58) : dans le cas de nouveaux produits, la réduction de leur prix grâce aux économies d'échelle peut susciter un accroissement de la demande, qui à son tour entraîne une augmentation de la taille

²⁸ Pour des raisons que nous développerons au chapitre VII, la possibilité d'acquérir à l'extérieur des composants logiciels (existence de marchés de composants logiciels), si elle est régulièrement annoncée, est restée très largement embryonnaire.

optimale, jusqu'à ce que celle-ci atteigne les limites fixées par la technologie employée (idem).

Ces deux phénomènes jouent pleinement dans le cas des progiciels. D'un point de vue technologique les économies d'échelle sont potentiellement infinies, et la faiblesse des coûts de transports et de communications (qui tendent vers zéro avec la possibilité de télécharger les logiciels) liée à un cercle vertueux de baisse des prix et d'extension des marchés, fait que la taille optimale d'une entreprise pour un type de progiciel et un type de matériel donnés, augmente très rapidement²⁹. De ce fait, l'effet *feedback* positif, qui "fortifie les forts et affaiblit les faibles" (Carl Shapiro, Hal R. Varian, 1999, p. 158) apparaît sans limite à la différence des industries traditionnelles. En effet, dans le cas général, le *feedback* positif dû aux économies d'échelle liées à l'offre, cesse d'exercer ses effets avec l'épuisement de celles-ci, et il est relayé par un effet *feedback* négatif, où "la force engendre la faiblesse et la faiblesse engendre la force" (idem) en raison des difficultés à gérer d'énormes organisations. Il faut également prendre en compte un phénomène particulièrement important dans le cas des progiciels (et plus généralement des biens d'information) : le rôle majeur des économies d'échelle liées à la demande ou effets de réseaux ou externalités de réseaux (Carl Shapiro, Hal R. Varian, 1999, p. 156-157). Ces économies d'échelle, à la différence de celles qui sont liées à l'offre, ne font qu'augmenter avec l'accroissement de la taille du marché. La combinaison d'économies d'échelle liées à l'offre *illimitées* et d'économies d'échelle liées à la demande, génèrent la poursuite de l'effet *feedback* positif jusqu'à la situation extrême de la domination d'un marché par une seule entreprise.

Le secteur des progiciels correspond à une situation de "monopole naturel" au moins pour un type technique donné de progiciel. Cette situation est caractéristique des activités où le monopole est "la structure de marché correspondant au plus bas prix de revient unitaire possible pour l'offre d'un service de meilleure qualité possible" (Jean-Pierre Angelier, 1997, p. 64). Dans de telles situations caractérisées par une fonction de coût continuellement sous-additive (le coût unitaire de la production est plus faible quand il est réalisé par une entreprise

²⁹ Ajoutons que dans le cas des progiciels, il n'existe pas d'impossibilité physique à ce qu'une entreprise couvre la totalité d'un marché et que les investissements à réaliser sont largement indépendants du nombre de produits vendus.

que lorsque cette production est répartie entre deux entreprises), le jeu de la concurrence aboutit à terme à une situation de monopole (Godefroy Dang Nguyen, 1995, p. 306)³⁰.

L'évolution de l'économie des progiciels, où la production s'effectue d'emblée pour le marché mondial, a validé cette analyse. Il faut préciser que le champ des applications informatiques étant vaste et originellement très segmenté (par type de progiciels mais également de matériels), cette concentration s'est tout d'abord opérée par domaine et non globalement (Christian Genthon, Denis Phan, 1999, p. 178)³¹. Elle a tout d'abord touché la production de systèmes d'exploitation, où la demande du marché (possibilités d'enrichissement permanent pour répondre aux évolutions du matériel, garantie de pérennité de l'offreur notamment) rejoint et renforce la dynamique industrielle d'une activité où les développements sont particulièrement importants pour produire des systèmes fiables, solides et faciles d'emploi dans une gamme étendue de domaines d'utilisation (Gérard Dréan, 1996 A, p. 212). Cette concentration monopolistique a été particulièrement nette dans le cas des progiciels pour micro-ordinateurs : outre le cas bien connu des systèmes d'exploitation où Microsoft dispose d'une situation de quasi-monopole consolidée sans discontinuer depuis 1981³², les différentes applications voient la domination très nette d'une entreprise³³ : par exemple en 1988, WorPerfect détenait 67 % du marché mondial des traitements de texte, Lotus 1-2-3 68 % du marché des tableurs, dBase III 63 % du marché des systèmes de gestion de bases de données (Frédéric Dromby, 1999, p. 264)³⁴. Mais une forte concentration existe

³⁰ Ce qui ne signifie pas nécessairement que ce marché n'est plus soumis à la concurrence, du moins si les coûts d'entrée et de sortie de ce marché sont faibles (cf. la théorie des marchés contestables de William J. Baumol, 1982).

³¹ "La réalité est que les firmes de progiciels ne constituent pas un secteur unique, mais autant de secteurs que de domaines d'application (systèmes d'exploitation, gestion de réseaux, traitement de texte, tableurs, utilitaires...), ces nombreux secteurs formant un métier unique. La dispersion n'est ici qu'apparente et concerne le niveau du métier. Chaque secteur élémentaire obéit à une logique de concentration, ce qui explique les fortes rentabilités constatées" (Gérard Dréan, 1996 A, p. 114).

³² Au milieu des années quatre vingt dix, la situation semblait évoluer inexorablement vers un monopole total de Microsoft sur les systèmes d'exploitation de micro-ordinateurs ; c'est l'action de Microsoft elle-même qui a empêché cette évolution (par crainte de sanctions judiciaires ?) en soutenant financièrement (prise de participation au capital en 1997) Apple dont la part de marché était passé à moins de 4 % et qui accumulait les pertes. Plus récemment, le succès d'un logiciel libre (Linux) atténue quelque peu la domination de Microsoft.

³³ Ceci ne signifie pas que l'entreprise qui domine le marché a toujours été la même, ce que nous analyserons dans le chapitre suivant.

³⁴ Un des rares segments, les logiciels graphiques, où étaient essentiellement présents deux acteurs d'importance à peu près égale, Adobe et Aldus, a vu la fusion de ces deux entreprises.

également dans le secteur des progiciels pour grands et moyens systèmes, par exemple Oracle dans le domaine des bases de données, SAP dans les progiciels de gestion d'entreprise, Computer Associate pour les logiciels destinés aux grands systèmes d'IBM. Cependant, dans les cas où la concentration s'opère par des opérations de fusion-absorption, elle est parfois freinée par les actions du *Department of Justice* ou de la *Federal Trade Commission*. Ceux-ci, sur la base de la législation antitrust américaine, ont interdit ou fait modifier plusieurs projets : Microsoft / Intuit dans les logiciels de finances pour les ménages, Silicon Graphics / Alias / Wavefront dans les logiciels graphiques haut-de-gamme pour stations de travail, Computer Associates / Legent dans les logiciels utilitaires pour grands systèmes IBM, Cadence / CCT dans les logiciels de conception électronique assistée par ordinateur (Carl Shapiro, Hal R. Varian, 1999, p. 268).

3 - L'apparition permanente de nouvelles entreprises sur de nouveaux segments

Toutefois ces phénomènes de concentration se produisant dans chaque sous-secteur ont été compensés par l'apparition permanente de nouveaux segments dans le secteur des logiciels et des services informatiques, qui ont constitué des opportunités pour la création de nouvelles entreprises.

Ce fut tout d'abord le cas avec la facturation séparée des logiciels et du matériel (1969), puis l'essor de la mini-informatique qui entraîna la création de nombreuses SSII (Frits Prakke, 1988, p. 68-69), à côté d'un secteur du matériel beaucoup plus concentré. Ces sociétés, au départ très dépendantes des constructeurs, s'en sont progressivement émancipées avec "la fin des systèmes propriétaires" (Serge Bouchy, 1994, p. 274). Au fur et à mesure que des opérations de concentration se déroulent dans le secteur du logiciel sur mesure et des services informatiques, l'extension des champs d'application de l'informatique provoque la création de nouveaux sous-secteurs caractérisés par des barrières à l'entrée faibles (par exemple récemment les services liés aux réseaux), qui sont occupés au départ par de nombreuses PME spécialisées sur cette nouvelle activité. Ainsi durant les années quatre-vingt-dix en France, il se crée en moyenne 4 000 nouvelles entreprises chaque année dans le secteur Activités informatiques (NAF 72), alors que le nombre d'entreprises du secteur reste stable aux alentours de 21 000 (INSEE, Enquête annuelle entreprises, Les entreprises des services en 1997, p. 836).

Le secteur des progiciels s'est développé avec l'essor de la micro-informatique. En même temps qu'elle constituait un marché de masse pour les logiciels produits, la révolution

micro-informatique permettait à de petites entreprises indépendantes, voire à des individus isolés de développer des logiciels (Gérard Dréan, 1996 A, p. 207). Se crée une nouvelle industrie du logiciel faite de centaines d'entreprises nouvelles, qui sont toutes au départ très petites (souvent formées d'une poignée de programmeurs) et qui produisent chacune un logiciel particulier pour un type d'utilisation précis (Frederick P. Brooks, 1996, p. 241). Ici également, la concentration qui s'opère rapidement sur chacun des sous-marchés, s'accompagne de la naissance de nouvelles entreprises pour le développement de nouvelles applications originales (par exemple *Netscape* avec l'apparition de navigateurs sur Internet, ou *McAfee Associates* pour les logiciels antivirus³⁵, ou encore *Checkpoint Software* avec son produit *Firewall 1* pour la protection des réseaux internes).

Enfin, la création de nouvelles entreprises a été facilitée par une série d'évolutions (pratique de l'essaimage ou *spin-off*, fonds de capital risque³⁶) non spécifiques au secteur informatique, mais qui ont pris une ampleur particulière dans ces secteurs, fortement innovants et à croissance rapide.

4 - Vers une remise en cause de ce schéma ?

On peut résumer l'évolution de la structure de l'économie du logiciel par une concentration forte sur chaque segment parvenu à maturité, partiellement masquée par l'existence de nombreux segments relativement indépendants³⁷, avec lors de l'apparition d'un nouveau segment (nouveau service, nouvelle application ou nouveau type de matériel) la présence de nombreuses nouvelles entreprises dont une forte proportion disparaîtront. Ces évolutions sont conformes au modèle général de l'évolution d'une industrie de W. Abernathy

³⁵ L'histoire de McAfee Associates, créée en 1992 et qui en 1997 détenait la moitié du marché mondial des logiciels antivirus, est édifiante. La création de cette société est l'œuvre d'un ingénieur qui travaillait pour Lockheed. Ce spécialiste de la sécurité informatique avait développé un logiciel antivirus (VirusScan) en 1989, l'avait distribué en shareware, ce qui lui avait rapporté cinq millions de dollars dès la première année.

³⁶ Par exemple, le produit actuellement phare de Lotus, le logiciel de travail en groupe Notes, fut mis au point par la start-up Iris financée par Mitchell Kapor, co-fondateur de Lotus (Frédéric Dromby, 1999, p. 693).

³⁷ La mesure de la concentration dépend de la délimitation du marché pris en référence, ce qu'illustre les divergences d'appréciation des protagonistes des procès antitrust aux Etats-Unis (Godefroy Dang Nguyen, 1995, p. 257). Par exemple, lors des poursuites contre IBM (1968-1982), cette société en se basant sur une définition large du marché (tous matériels de bureaux) estimait qu'elle ne détenait que 38 % du marché, alors que l'Etat fédéral américain, en se limitant au marché des mainframes, considérait que sa part de marché était de 70 %. Il en est de même actuellement pour Microsoft, selon que l'on considère que le marché pertinent est celui des systèmes d'exploitation pour micro-ordinateurs, de l'ensemble des progiciels pour micro-ordinateurs, voire de l'ensemble des logiciels tous matériels confondus.

et J. Utterbach (1978), caractérisé par une première phase "d'entrée facile", suivi d'une phase de consolidation après l'émergence d'une conception dominante (*dominant design*). La particularité de l'industrie du logiciel serait la fréquence particulièrement rapide de création de nouveaux sous-marchés, "des sous-secteurs nouvellement créés [reprenant] le cycle à son début et [présentant] une structure très différente de celle du secteur dont ils sont issus" (Gérard Dréan, 1996 A, p. 115). En conséquence, "le logiciel reste un domaine particulièrement divers et mouvant" (idem, p. 255), caractérisé par les ascensions fulgurantes de certains acteurs et les chutes retentissantes d'autres entreprises, mais également par une relative stabilité de la proportion des firmes de chaque taille.

On peut toutefois se demander si *un certain nombre de transformations récentes ne risque pas de remettre en cause cette situation, la concentration dans l'ensemble du secteur du logiciel durablement l'apparition de nouvelles entreprises*. Selon Carl Shapiro et Hal R. Varian, pour les biens d'information, seules deux structures de marché sont viables à long terme : d'une part, "un marché contrôlé par une *entreprise dominante* qui ne produit pas forcément le meilleur produit, mais qui, par sa taille et grâce aux économies d'échelle dont elle bénéficie, écrase ses concurrents" ; d'autre part, "un marché de produits différenciés (...) structure sans doute la plus répandue" (1999, p. 30). Plusieurs facteurs (interreliés) rendent crédible une évolution tendancielle vers la première structure de marché :

- premièrement, une *atténuation des frontières entre les différents sous-segments*. Ce phénomène concerne les progiciels (par exemple le succès des "suites" bureautiques regroupant l'ensemble des applications bureautiques, ou des progiciels intégrés de gestion d'entreprise), les services et les logiciels sur mesure (recherche de solutions globales), les différents types de matériel avec la fusion tendancielle en un seul réseau technico-économique de l'ensemble de l'informatique (cf. chapitre II). Par exemple, au niveau des systèmes d'exploitation, où auparavant existaient des marchés strictement séparés entre systèmes d'exploitation pour mainframes, mini-ordinateurs, stations de travail et micro-ordinateurs, la concurrence va s'exercer de plus en plus sur l'ensemble de l'informatique : développement de versions de Windows, qui était initialement positionné sur la micro-informatique, vers les serveurs (Windows NT, Windows 2000), extension des différentes "familles" d'Unix des stations de travail vers les mainframes et les micro-ordinateurs...

- deuxièmement, *la nécessité d'investissements de départ plus importants* tant dans les logiciels sur mesure et les services informatiques (proportion croissante de coûts fixes) que

dans des progiciels de plus en plus volumineux³⁸. Or, pour les progiciels, si les coûts variables, qui sont des coûts de reproduction et qui sont très faibles, sont à peu près proportionnels à la taille du progiciel, les coûts fixes de développement croissent plus rapidement que la taille du progiciel, notamment en raison des dépenses de tests et d'intégration dont la productivité baisse fortement quand la taille augmente (cf. chapitre V). A l'élévation de ces barrières à l'entrée de nature technique, s'ajoutent des barrières à l'entrée "stratégiques" qui résultent des comportements des entreprises en place (par exemple, l'explosion des dépenses de communication pour lancer un progiciel sur le marché).

- troisièmement, *l'existence d'acteurs à la puissance financière considérable* et, qui ayant tiré les leçons des expériences précédentes, réagissent de plus en plus rapidement face à l'apparition d'un nouveau segment, soit en tentant de s'y développer directement, soit en rachetant précipitamment les nouveaux acteurs qui apparaissent (l'exemple caricatural étant le comportement de Microsoft).

Ces acteurs peuvent avoir une activité principale autre que la production des logiciels, ce qui conduit à examiner la diversité des producteurs selon une troisième dimension concernant leur plus ou moins grande spécialisation dans le développement de logiciels.

C - EN TERMES D'ACTIVITE PRINCIPALE DES PRODUCTEURS : DES TENDANCES APPAREMMENT CONTRADICTOIRES A LA SPECIALISATION ET A LA DIVERSIFICATION

Il ne s'agit pas ici de traiter de la question traitée précédemment (cf. A) des entreprises qui produisent des logiciels pour leurs besoins internes, mais de l'activité principale des entreprises qui commercialisent des logiciels. En effet, coexistent des entreprises dont c'est l'activité quasi-exclusive, avec des entreprises dont l'activité principale est autre (producteurs de matériel informatique, opérateurs de télécommunications, sociétés de conseil, groupes bancaires ou industriels). Cette situation est le produit de la désintégration et de la

³⁸ La première version de dBase (produit longtemps leader pour les systèmes de gestion de bases de données sur micro-ordinateur) fut développée par une seule personne, Wayne Ratliff à la fin des années soixante-dix en dehors de ces heures de travail, alors qu'une version ultérieure de ce produit (dBase IV créé dix ans plus tard) a mobilisé 150 programmeurs pendant trois ans (Frédéric Dromby, 1999, p. 636 et 640).

spécialisation qui se sont opérées dans l'industrie informatique initialement intégrée (1) et, plus récemment des stratégies de diversification de nombreuses entreprises (2).

1 - Une tendance forte à la désintégration et à la spécialisation

a - L'intégration initiale de l'industrie informatique

L'industrie informatique se compose de "l'ensemble des organisations qui produisent des biens ou des services entrant dans la composition des systèmes informatiques ou utilisés dans leur mise en œuvre" (Gérard Dréan, 1996 A, p. 108), les systèmes informatiques étant constitués par des matériels interconnectés, sur lesquels s'exécutent des logiciels qui en définissent les fonctions applicatives. L'informatique est au départ une activité secondaire des firmes mécanographiques et des grandes firmes d'électronique, et elle s'organise comme industrie selon un modèle vertical, fondé sur une intégration forte autour d'un petit nombre de firmes globales, dont l'archétype est IBM (Claire Charbit, Jean-Benoît Zimmermann, 1997, p. 1). Jusqu'aux années quatre-vingt, il semble évident que les seules entreprises qui peuvent prétendre jouer un rôle important dans ce secteur – ou même simplement survivre – sont les entreprises généralistes présentes sur l'ensemble de la filière informatique. Cette intégration semblait devoir même s'étendre à l'ensemble de l'électronique voire aux télécommunications³⁹.

En réalité, l'évolution de l'industrie informatique s'est effectuée conformément à la théorie développée par George Stigler (1951). Selon cet auteur, une industrie nouvelle apparaît nécessairement sous une forme intégrée, mais le développement du marché favorise une spécialisation horizontale des entreprises et une désintégration verticale de l'industrie⁴⁰. S'il demeure dans l'informatique des entreprises généralistes dont l'activité s'étend depuis la production des composants jusqu'à la fourniture des logiciels et des services, leur importance

³⁹ Le pronostic le plus fréquent était que la convergence entre l'informatique et les télécommunications conduirait ces industries à se structurer principalement autour d'un duel titanesque entre ATT et IBM. Malgré la réalité technologique de cette convergence autour de la numérisation, ces deux industries sont restées économiquement séparées avec des structures différentes, et il est significatif de voir à l'inverse IBM et ATT conclure un accord récent, au terme duquel IBM cède ses activités de télécommunications à ATT, et ATT ses activités informatiques à IBM.

⁴⁰ Dans le cas de l'informatique il n'est évidemment pas possible de vérifier si, comme l'énonce G. Stigler, lors de la phase de stagnation et de déclin de l'industrie se produit un processus de réintégration progressive des entreprises restantes.

relative a considérablement déchu au profit d'entreprises spécialisées⁴¹. Lucio Stanca, dirigeant du "seul acteur occidental présent sur l'intégralité du spectre des métiers de l'informatique, IBM" (Eurostaf, 1995 A, p. 37) résume ainsi ce changement : "dans les années soixante-dix, nous n'avions que quelques concurrents bien identifiés. L'industrie informatique s'est ensuite désintégrée, avec la naissance d'une prolifération de concurrents très spécialisés sur un créneau très précis. La concurrence est devenue la plus compétitive de tous les secteurs économiques, et nous pensons aujourd'hui être confrontés à plus de 50 000 entreprises concurrentes d'une manière ou d'une autre" (cité in Eurostaf, 1996 A, p. 137).

b - Désintégration verticale et spécialisation progressive

Il s'est tout d'abord opéré une première spécialisation correspondant à l'apparition de nouveaux acteurs autour de la constitution de plusieurs réseaux technico-économiques dans l'informatique (cf. chapitre II). Dans certains cas, ces entreprises si elles étaient spécialisées dans une informatique particulière, sont restées relativement intégrées verticalement (généralistes sur un segment de l'informatique) : par exemple DEC pour la mini informatique, SUN pour les stations de travail ou Apple pour la micro-informatique ont toujours eu une activité de producteurs de matériels et de logiciels.

La spécialisation sur un stade particulier de la filière informatique (composants, assemblage de matériel, périphériques, logiciels, services...) qui dérive de "la nature, par essence combinatoire, du produit informatique" (Claire Charbit, Jean-Benoît Zimmermann, 1997, p. 1) n'a pu s'approfondir qu'à partir de l'existence de standards et de la fin des systèmes "propriétaires"⁴². C'est pourquoi si une relative désintégration s'effectue précocement par nature de prestations - à la suite de la facturation séparée des services, puis plus généralement entre le *hard* et le *soft* (David C. Mowery, 1996, p. 308) - elle s'accroît dans les années quatre-vingt avec l'essor de la micro-informatique, dont la structure est d'emblée constituée, à l'exception d'Apple, d'un grand nombre de spécialistes offrant chacun un élément du système. Cette structuration de l'industrie micro-informatique est un des facteurs explicatifs de déclenchement du cercle vertueux de baisse des prix et de diffusion de la micro-informatique

⁴¹ Selon Jean-Pierre Brulé (1993, p. 295), les entreprises généralistes représentaient 90 % du marché informatique il y a trente ans, 50 % aujourd'hui et cette part ne dépasserait pas 40 % d'ici trois ou quatre ans.

⁴² "Lorsqu'un standard est adopté, la concurrence se joue entre les composants et non plus entre les systèmes (...) La standardisation fait la prospérité des fabricants d'équipements spécialisés. Inversement l'incompatibilité des systèmes fait l'affaire des généralistes" (Carl Shapiro, Hal R. Varian, 1999, p. 206).

qui aboutit à une inversion radicale de la "loi de Grosh" qui énonçait qu'à coût égal un grand système informatique offrait plus de puissance que plusieurs petits systèmes. Cet éclatement de l'industrie en secteurs spécialisés s'étend progressivement du monde des PC à l'ensemble de l'informatique (Gérard Dréan, 1996 A, p. 286).

Les facteurs de la spécialisation

Les avantages des entreprises spécialisées proviennent des caractéristiques très différentes des activités qui composent l'industrie informatique en termes d'échelle optimale de production, de durée du cycle de vie des produits, de compétences nécessaires, d'exigences organisationnelles et de dynamique concurrentielle... De ce fait, une entreprise spécialisée pourra adapter sa culture ou sa personnalité aux caractéristiques de l'activité et être plus compétitive. Une entreprise généraliste a la possibilité de compenser par des profits plus importants sur les segments d'activité où la concurrence est plus faible, des pertes sur certaines activités soit parce qu'elle y est moins efficace, soit en raison de sa stratégie commerciale (prix bas pour conquérir la marché, pouvant aller jusqu'à la fourniture gratuite de certaines prestations). Mais l'existence de segments à profitabilité élevée risque d'attirer de nouveaux producteurs spécialisés dans cette activité, ce qui entraîne une recherche de profits plus élevés pour les entreprises généralistes sur les segments qui restent protégés, augmentant l'intérêt pour de nouvelles firmes spécialisées de tenter de pénétrer ces derniers segments protégés : "plus les segments attaqués par les spécialistes se multiplient, plus les segments relativement protégés se raréfient et deviennent vulnérables. (...) L'éclatement de l'offre des généralistes et le foisonnement des spécialistes se nourrissent donc l'un de l'autre et gagnent de proche en proche la totalité du spectre des produits et des services" (Gérard Dréan, 1996 A, p. 293).

De plus, l'élévation de la taille optimale et l'importance des investissements à réaliser nécessitent d'opérer des choix en présence de ressources financières qui sont plus limitées avec la sortie de l'ère "propriétaire" (Eurostaf, 1995 A, p. 37). Enfin, l'importance des innovations et la difficulté à couvrir l'ensemble des évolutions technologiques⁴³ poussent les entreprises à se centrer sur un métier stratégique, où leur excellence technologique peut leur

⁴³ Il faut également ajouter que dans le cas d'une entreprise généraliste, la probabilité est plus forte qu'une innovation ne soit pas introduite car concurrençant directement une autre activité de l'entreprise. Ce phénomène fut particulièrement marquant à IBM.

permettre d'apparaître comme des acteurs majeurs voire incontournables de l'activité (Serge Bouchy, 1994, p. 276 et 314).

Un des freins à la spécialisation dans l'industrie informatique résidait dans le manque de compétences des entreprises clientes pour s'adresser à des fournisseurs distincts et dans leurs préférences pour des solutions globales, dans un domaine où les relations d'un produit avec les autres produits ont souvent autant d'importance que les caractéristiques propres de chaque produit. Toutefois ce problème tend à être résolu par l'augmentation des compétences des utilisateurs et par l'apparition d'une nouvelle activité, l'intégration de systèmes, où des sociétés de services spécialisées se sont implantées, moins suspectes aux yeux des entreprises clientes que les constructeurs, de chercher à vendre abusivement du matériel (Jean-Marie Desaintquentin, Bernard Sauter, 1991, p. 48).

Les avantages des entreprises spécialisées sur les entreprises généralistes sont confirmés par l'évolution des performances des deux types de firmes. Alors que les grands constructeurs généralistes connaissent des difficultés (baisse de leur volume d'affaires, pertes financières) qui leur imposent de douloureuses restructurations (suppression d'emplois, retrait partiel ou total de certaines activités), des entreprises spécialisées connaissent des taux de croissance de 30 % et plus, et des marges nettes supérieures à 20 % (Gérard Dréan, 1996 A, p. 77). Ces entreprises spécialisées sont le plus souvent de nouvelles entreprises qui se sont d'emblée centrées sur un créneau (Cray pour les supercalculateurs par exemple) ou sur une activité (Intel pour les microprocesseurs, Microsoft pour les logiciels). Pour certaines activités comme les services informatiques et les logiciels sur mesure, ce processus peut résulter d'une filialisation des services informatiques internes de grands groupes industriels ou de services. Si, dans un premier temps, ces filiales réalisent l'essentiel de leur activité pour leur entreprise d'origine, il est fréquent qu'elles élargissent leur clientèle à l'ensemble du marché (par exemple en France, Dassault Systèmes et Matra Datavision qui développent et commercialisent des logiciels de conception et de fabrication assistées par ordinateur) allant parfois jusqu'à devenir complètement indépendantes⁴⁴.

⁴⁴ L'exemple le plus célèbre est la première société de services informatique au niveau mondial EDS : filiale de General Motors, sa part de chiffre d'affaires réalisée auprès de GM est passée de 53 % en 1990, à 47 % en 1991, et à 39 % en 1993 (Eurostaf, 1994 A, p. 262). En février 1996, elle est devenue indépendante de General Motors.

Les caractéristiques profondément différentes des diverses activités qui constituent l'informatique sont illustrées par les difficultés rencontrées par des entreprises dont la culture et l'organisation étaient bien adaptées à un secteur informatique, pour s'imposer dans un autre secteur. Par exemple, alors que les progiciels et les services informatiques sont souvent abusivement regroupés dans un même secteur, aucune SSII n'a réussi, malgré les efforts de certaines d'entre elles (EDS, Sema Group...), à être un producteur significatif de progiciels (Eurostaf, 1996 A, p. 116 et 227). De même, si IBM a su utiliser sa puissance en tant que constructeur dans le domaine des logiciels de base pour grands et moyens systèmes informatiques, elle a accumulé les échecs dans les logiciels d'application et dans les systèmes d'exploitation pour micro-ordinateurs. La seule véritable réussite d'un constructeur d'ordinateurs dans le domaine des services informatiques ou des logiciels est celle de Wang, mais son succès sur certains marchés (*workflow*, *groupware*, intégration) a nécessité un changement du métier principal de l'entreprise qui a abandonné sa production initiale de mini-ordinateurs (Eurostaf, 1995 A, p. 37).

Une spécialisation de plus en plus fine

La spécialisation des entreprises ne s'est pas limitée à l'exercice d'une seule des activités constitutives de l'informatique. Notamment dans le cas des logiciels et des services informatiques, elle a pris la forme d'une spécialisation encore plus fine sur un seul type de prestation. La diversification croissante des applications de l'informatique et des attentes des utilisateurs (cf. section I) crée de nouvelles opportunités de spécialisation (Géard Dréan, 1996 A, p. 115-116). Dans les services informatiques, la nature même de l'activité met l'ensemble de l'entreprise au contact direct du marché, ce qui fait que l'adéquation de la personnalité de l'entreprise aux attentes du marché est un élément-clé de ses performances. Les services informatiques (conseil, assistance, formation, développement de logiciel, ingénierie de système, intégration de système...) présentent entre eux suffisamment de différences "pour exiger des structures, des systèmes de management et des cultures d'entreprises différentes" (idem, p. 268-269). De fait, les sociétés de services étroitement spécialisées semblent obtenir de meilleurs résultats que les SSII multi-secteurs et multi-activités (Abdelaziz Mouline, 1996, p. 30). Selon Eurostaf (1996 A, 1997 A), les difficultés actuelles des SSII françaises s'expliqueraient par leur caractère trop "généralistes" et elles devraient se centrer plus précisément sur une activité, seule possibilité pour industrialiser leurs prestations, atteindre la taille critique et obtenir une reconnaissance internationale dans cette activité.

Dans le secteur des progiciels, la plupart des entreprises sont spécialisées sur un type de matériel (par exemple Computers Associates pour les mainframes) et/ou sur une activité précise : les progiciels de gestion d'entreprise pour SAP, les bases de données pour Oracle, les logiciels graphiques pour Adobe et Aldus, les logiciels de réseaux pour Novell, les logiciels de conception assistée par ordinateur pour Autodesk, les logiciels de sécurité informatique pour Network Associates. Dans les progiciels verticaux destinés à une profession spécifique, il existe également fréquemment une spécialisation des entreprises sur un métier particulier, qui ne peut que se renforcer avec l'importance croissante du domaine d'application au détriment des compétences informatiques générales⁴⁵. Ici également, c'est dans la micro-informatique que la spécialisation est poussée à l'extrême, certaines entreprises de progiciels pour micro-ordinateurs réalisant l'essentiel de leur chiffre d'affaires par la vente d'un seul produit : par exemple, en 1988, Microsoft réalisait la moitié de son chiffre d'affaire avec MSDOS, Lotus les deux tiers avec son tableur 1.2.3 (les trois quarts en 90 et 91), Ashton Tate les deux tiers avec son système de gestion de bases de données dBase (et 100 % de son bénéfice), WordPerfect les deux tiers avec son traitement de texte, et Software Publishing Corporation les deux tiers également avec son logiciel de présentation Harvard Graphics (Frédéric Dromby, 1999, p. 181-182). On peut ajouter qu'une entreprise présente sur un segment a rarement réussi sur un autre segment à l'exception notable de Microsoft (David C. Mowery, 1996, p. 11).

2 - Une tendance apparemment contradictoire de recherche de diversification des entreprises

Les avantages des entreprises spécialisées ne signifient pas qu'il n'existe que ce type d'entreprise dans le secteur informatique et notamment dans l'économie du logiciel. Les tentatives de diversification, l'attrait que représentent des activités où les profits peuvent être très élevés, ont également toujours existé et se sont même renforcés dans la période récente.

Cette diversification peut être de nature conglomerale lorsqu'une entreprise s'engage dans une activité n'ayant aucun lien direct avec son activité initiale, en fonction de certains critères, comme par exemple le taux de croissance du marché et la part du marché détenue par la firme (méthode du *Boston Consulting Group*). La diversification est de nature concentrique

⁴⁵ "La manière de faire quelque chose dépend bien davantage du segment que de l'utilisation de méthodes générales d'analyse des systèmes, de langages généraux, de techniques de tests générales" (Tom DeMarco, cité in Frederick P. Brooks, 1996, p. 245).

lorsque l'activité nouvelle a un rapport technique ou commercial avec le métier de base de la firme. Dans ce cas, l'engagement dans des productions qui sont complémentaires des technologies ou des marchés existant dans le cadre de l'entreprise, est motivé par la possibilité d'économies d'association (ou économies d'envergure ou encore économies de variété).

La diversification vers les logiciels d'entreprises extérieures à ce secteur

C'est ce qui explique que des entreprises situées dans des secteurs extérieurs à l'informatique mais où il existe des synergies de contenu cognitif, aient investi le champ des logiciels et des services informatiques. C'est notamment le cas des sociétés de conseil et d'audit et des opérateurs de télécommunications. Le secteur du conseil et de l'audit connaît un ralentissement de sa croissance et une érosion de ses frontières avec une partie des services informatiques⁴⁶. Cette situation pousse les principales entreprises du secteur à élargir leur activité dans l'informatique à partir de prestations de conseil en informatique - dont la croissance est deux fois supérieure à celle du conseil traditionnel (Eurostaf, 1995 B p. 43) - pour capitaliser leur portefeuille de clients en répondant à des besoins de prestations plus globales. Les opérateurs de télécommunications, confrontés à des évolutions techniques (numérisation et nouveaux services) et économiques (passage d'un marché d'offres à un marché d'utilisateurs avec la déréglementation du secteur, la libéralisation des marchés et la fin des monopoles), veulent contrôler les multiples logiciels (gestion des communications, des clients, des tarifs...) qui deviennent une arme décisive de la nouvelle concurrence dans ce secteur. C'est également le cas d'entreprises produisant des biens d'équipement sophistiqués dans lesquels les logiciels ont une place de plus en plus importante : par exemple, dans le secteur des équipements médicaux, il existe une très forte concurrence par les prix entre les entreprises, qui réalisent l'essentiel de leur profit par la vente des logiciels complémentaires (Carl Shapiro, Hal R. Varian, 1999, p. 146).

Mais c'est évidemment avec les entreprises du secteur informatique que la complémentarité des activités est la plus importante, qu'il s'agisse de nouveaux acteurs (par exemple AOL absorbant Netscape) ou d'acteurs traditionnels comme les constructeurs de grands et de moyens systèmes qui sont tous présents sur les marchés des logiciels et des services informatiques. Ce positionnement constitue un revirement par rapport à l'attitude des

⁴⁶ Selon Bossard Consultants, les technologies de l'information représentent 39 % du marché européen du conseil en gestion (Eurostaf, 1996 A, p. 169).

constructeurs qui dans les années soixante-dix et quatre-vingt s'étaient désengagés au profit principalement des SSII, par la facturation séparée de ces activités qu'elles fournissaient au départ gratuitement. Cette évolution correspond à une tendance plus générale des firmes de haute technologie à produire simultanément des biens d'équipement et des services (Faïz Gallouj, 1994, p. 99). Elle est liée à un déplacement du marché, initialement centré sur le matériel, vers les services et le logiciel (Jean-Marie Desaintquentin, Bernard Sauter, 1991, p. 47). Une motivation importante pour certains constructeurs peut être de tirer un revenu de ressources en personnel devenues excédentaires (Gérard Dréan, 1996 A, p. 304). Il peut s'agir également d'une stratégie pour s'implanter à l'intérieur des entreprises et faciliter la vente de matériels dans un environnement devenu plus concurrentiel avec la fin des systèmes "propriétaires" (Eurostaf, 1997 A, p. 85). L'érosion des marges bénéficiaires dans le domaine du matériel, qui résulte de l'exacerbation de la concurrence, incite les constructeurs disposant d'une infrastructure commerciale et de moyens financiers importants, à se tourner vers les activités de logiciels et de services réputées plus profitables.

Il faut toutefois tenir compte du fait, que, dans les cas fréquents où la vente du produit principal de l'entreprise s'appuie sur des relations de partenariat avec des entreprises commercialisant des produits complémentaires, l'entrée sur ces marchés risque de remettre en cause les accords passés et peut donc affaiblir les ventes du produit principal. On peut également remarquer que c'est seulement dans le cas des logiciels que l'on rencontre des profits particulièrement élevés, le secteur des services informatiques n'ayant pas un niveau de profit supérieur à celui du matériel, même si certains segments ont des taux de croissance élevés (Gérard Dréan, 1996 A, p. 258). C'est pourtant essentiellement vers les services informatiques que les constructeurs informatiques vont se diversifier, non sans difficultés pour certains d'entre eux⁴⁷. Dans les services informatiques, la part des constructeurs augmente significativement durant les années quatre-vingt-dix : par exemple en France, elle passe de 4,1 % en 1991 à 19 % en 1995 du marché des services informatiques réalisé par les trente premiers acteurs (Eurostaf, 1996 A, p. 237) et à 33,6 % en 1996 du marché réalisé par les cinquante premiers prestataires (Eurostaf, 1997 B, p. 252). L'évolution d'IBM est éloquent : depuis 1995, la part du logiciel et des services a dépassé le matériel dans les

⁴⁷ Sur les difficultés en termes de compétences nécessaires, de mode de management, de changement culturel qu'implique une telle évolution, cf. pour Olivetti, Christian Genthon (1995, p. 145) et pour Bull, Jean-Pierre Brulé (1993, p. 249). Pour Bull la branche « conseil et intégration de systèmes », a perdu 41 millions d'euros en 1999 (les pertes totales de Bull étant de 288 millions d'euros en 1999).

ventes de cette société avec pour objectif les deux tiers du CA total pour la fin de la décennie (Eurostaf, 1996 A, p. 136). Le regroupement de toutes les activités de services d'IBM dans *IBM Global Services* le premier janvier 1997 ont fait de cette entité le premier acteur des services informatiques sur le marché mondial comme sur le marché européen ou français. Par contre, pour les progiciels l'évolution est plus difficile et c'est par une OPA hostile sur Lotus en 1995 (la première de l'histoire d'IBM depuis quatre-vingts ans), représentant à l'époque la plus grosse transaction jamais effectuée dans l'industrie du logiciel (17,5 milliards de F), qu'IBM tente de s'imposer dans ce secteur.

La diversification des entreprises productrices de logiciels

La diversification des entreprises ne s'opère pas uniquement vers le secteur des logiciels et des services informatiques. En sens inverse, certaines entreprises de ce secteur tentent de se diversifier dans des activités complémentaires. Par exemple, le recouvrement des marchés du conseil et des services amène également les acteurs des services informatiques à étendre leurs prestations en amont vers le conseil (rachats par EDS de la société américaine de consultants AT Kearney et de la société française de conseil en management Eurosept, rachats par Cap Gemini de United Research, de Gruber Titze & Partner, de Bossard Consultants et dernièrement de Ernst & Young). Si, notamment dans les domaines nouveaux, des entreprises très spécialisées sur un progiciel ou une prestation particulière connaissent des succès indéniables, leur position peut se révéler très fragile lorsque se produisent des évolutions technologiques ou économiques (implantation sur ce marché d'entreprises plus puissantes), comme l'a démontré la disparition d'entreprises leaders sur certains segments de la micro-informatique (CPM, WordPerfect, Aston Thate...). C'est pourquoi les entreprises spécialisées sur un type de produit tentent de se diversifier sur des produits voisins. Par exemple, les entreprises spécialisées dans les progiciels micro-informatiques essaient d'élargir leur marché vers le reste de l'informatique (Microsoft avec Windows NT pour les serveurs et les stations de travail, ou Adobe qui a racheté en 1995 Frame Technology, entreprise qui développe essentiellement des logiciels pour Unix)⁴⁸. Cette évolution facilitée par les convergences technologiques et économiques au sein de l'ensemble de l'informatique (cf. chapitre II) est identique à celle à l'œuvre dans le domaine du matériel, où on voit certains constructeurs de

⁴⁸ En sens inverse, des firmes spécialisées dans les logiciels pour grands systèmes développent leur offre en direction des systèmes sous Unix ou des micro-ordinateurs, par exemple Compuware, BMC Software, SAS Institute (Eurostaf, 1996 A, p. 121).

micro-ordinateurs, à partir de leur succès dans la micro-informatique, s'attaquer aux autres marchés du matériel informatique, où les marges sont plus élevées (par exemple Compaq, un des premiers producteurs de micro-ordinateurs, a racheté Tandem, producteur de *mainframes* en 1997 et Digital Equipment, producteur de mini-ordinateurs en 1998).

Pour les entreprises qui ont acquis une puissance financière importante grâce à leur domination dans le domaine des progiciels, cette diversification peut déborder le cadre des logiciels, voire de l'informatique. C'est le cas de Microsoft qui, outre sa diversification au sein du secteur des logiciels⁴⁹, notamment par le rachat de multiples entreprises⁵⁰ (l'éditeur de logiciel de création graphique Visio, la société canadienne Softimage spécialisée dans les logiciels d'animation et visualisation graphique en trois dimensions, la société américaine Render Morphics spécialisée dans les logiciels graphiques multimédias trois dimensions...), multiplie les tentatives d'implantation sur les marchés complémentaires de son activité principale : les services informatiques, les "contenus" (encyclopédie, cinéma, presse...)⁵¹, les sites Web (rachat de Hotmail qui drainait un important trafic en proposant l'hébergement gratuit de boîtes aux lettres), les satellites (Teledisc), les câblo-opérateurs (Comcast), la télévision sur Internet (WebTV), les consoles de jeux....

Les accords inter-entreprises

Une forme particulière des politiques de diversification réside dans la conclusion d'accords inter-entreprises qui se multiplient dans le secteur des technologies de l'information et de la communication. Ces accords recouvrent une grande variété de formes contractuelles

⁴⁹ Outre ses tentatives pour s'implanter sur les marchés des progiciels pour des systèmes plus puissants que les micro-ordinateurs, Microsoft essaye également de s'imposer avec Windows CE, pour le moment avec un succès limité, sur le marché en plein développement des systèmes d'exploitation pour l'ensemble des objets qui contiennent des microprocesseurs (les téléphones, les décodeurs, les consoles de jeux, les multiples cartes à puces, les agendas électroniques, les ordinateurs de voiture...).

⁵⁰ Selon Roberto Di Cosmo et Dominique Nora, Microsoft prend des participations dans trente PME technologiques chaque année (1998, p. 107).

⁵¹ La réussite la plus spectaculaire parmi de multiples initiatives (la chaîne d'informations MSNBC, la banque d'images et d'œuvres numériques Corbis...) est la place acquise par Microsoft dans le domaine des encyclopédies, au détriment notamment de Britannica, en créant une encyclopédie sur CD-ROM à partir du rachat des droits de Funk & Wagnalls, encyclopédie jusqu'alors vendue dans les supermarchés, puis en y ajoutant le contenu de l'encyclopédie Collier. Cette évolution vers les "contenus" que l'on retrouve également pour d'autres producteurs de progiciels (Corel a racheté GraphicCorp, spécialisée dans les "contenus" numériques : graphismes, photographies, images clipart, animations Web, applications téléchargeables, Lotus à l'époque numéro un des tableurs avait racheté en 1987 une société spécialisée dans l'information financière) n'est pas un phénomène nouveau dans l'informatique : dans les années quatre-vingt les SSII avaient tenté de s'implanter dans le secteur des banques de données.

et organisationnelles : accords de licence, prises de participation partielle, joint-ventures, rachats, fusions, constitutions de consortium... Abdelaziz Mouline définit un accord comme l'établissement de relations entre firmes "à long ou moyen terme pour partager un ensemble limité de ressources de diverses natures (moyens financiers, équipements, technologies, etc.) sans pour autant remettre en cause l'autonomie des firmes en question" (1996, p. 55). L'existence d'accords entre entreprises est un phénomène qui s'est développé en même temps que la spécialisation croissante des entreprises en réponse à la recherche de solutions globales de la part des organisations utilisatrices : partenariats entre constructeurs informatiques et SSII notamment dans les grands et moyens systèmes⁵², entre éditeurs de progiciels, sociétés de services et cabinets de conseils (par exemple les accords autour des progiciels de gestion de SAP, ou le programme Partenaires Solutions lancé en septembre 93 par Microsoft autour de Windows NT). Ces alliances entre entreprises présentent l'avantage pour les producteurs de fidéliser leurs clients par l'élargissement d'une offre de produits complémentaires. Pour les entreprises utilisatrices, elles permettent de traiter avec un interlocuteur unique ("intégrateur" de systèmes), ce qui autorise une meilleure gestion de l'hétérogénéité, et d'obtenir des engagements de résultats et de délais, tout en maîtrisant les coûts grâce aux recours aux compétences spécialisées des différents partenaires.

Les accords se sont multipliés et élargis avec la montée en puissance du multimédia et plus récemment du commerce électronique. Le multimédia est la réunion sur un même support d'informations numérisées représentatives de sons, de données et d'images – animées ou pas – combinées selon un schéma intellectuel ou artistique original, et dont la consultation (*off line* ou *on-line*) implique l'utilisation d'un logiciel de navigation permettant une interactivité entre l'homme et le produit. Il résulte de la convergence, autour de la numérisation de l'information, des technologies de l'audiovisuel, de l'informatique et des télécommunications. Le développement du multimédia nécessite des investissements très élevés et la réunion de compétences très diverses détenues par des acteurs différents (détenteurs de contenus⁵³, fabricants de matériels informatiques, de semi-conducteurs, d'électronique grand public, producteurs de logiciels, opérateurs de télécommunications,

⁵² A elle seule, IBM revendique 300 accords avec participation au capital et 20 000 business partnership, qui sont cependant en majorité de simples accords de distribution (Christian Genthon, 1995, p. 110).

⁵³ Cf. l'étude du groupe de travail sur l'économie de l'information de l'OCDE intitulée significativement "Les contenus : un nouveau secteur de croissance" (OCDE, 1998 B).

câblo-opérateurs, sociétés de services...). L'existence de très forts rendements croissants d'adoption et la possibilité de valoriser un même prototype sur des supports de plus en plus diversifiés encouragent de multiples acteurs à se positionner rapidement sur le marché du multimédia promis à un développement accéléré. Ce positionnement peut s'opérer par des rachats et fusions aboutissant à la création "d'entreprises géantes disposant des équipements et des logiciels nécessaires pour contrôler entièrement messages et images, depuis le stade de la conception jusqu'à celui de la mise à disposition finale aux utilisateurs ou au grand public" (Herbert I. Schiller, 1997, p. 47-48). La récente fusion entre AOL et Time Warner, dont il est prématuré d'évaluer les effets, illustre cette possibilité. Toutefois, il n'est nullement évident que cette stratégie soit la plus efficace, dans la mesure où il est peu probable que la convergence technique entre les différentes industries élimine les frontières entre celles-ci (Christian Genthon, 1995, p. 165). C'est ce que semble confirmer une étude d'Abdelaziz Mouline, qui montre que si les opérations de restructuration dans le multimédia ont très fortement augmenté dans les années quatre-vingt-dix, les alliances (principalement des alliances intersectorielles de Recherche-Développement et de commercialisation) sont la modalité privilégiée par les différents acteurs du multimédia (80 % des opérations) au détriment des opérations de rachats et de fusions qui ne représentent que 20 % (1996, p. 87).

De façon plus générale, Gérard Dréan estime que la plupart des acquisitions sont des échecs, surtout si les entreprises opèrent dans des secteurs différents, notamment à cause de la sous-estimation des problèmes réels de compatibilité entre cultures (1996 A, p. 324), particulièrement décisifs dans des activités où prédominent les actifs immatériels. Concernant les accords entre entreprises, il importe de distinguer la réalité de l'accord, c'est à dire les dispositions concrètes effectivement appliquées, de sa présentation où les effets d'annonce et les motivations d'image sont souvent dominantes. En réalité, une grande partie des accords a peu d'incidence sur l'activité des différents partenaires : selon une étude de Mac Kinsey, les deux tiers des accords sont improductifs (Christian Genthon, 1995, p. 111). De plus, dans une activité où les évolutions sont extrêmement rapides, les accords peuvent être périmés avant d'avoir un début d'application significatif, voire même comporter "une dimension imaginaire, permettant de conjurer un futur de moins en moins prévisible" (idem). Enfin, les alliances les plus efficaces sont celles qui ont un objectif limité et précis, de préférence aux alliances trop ambitieuses ou trop vagues qui ont toutes chances d'échouer, sans parler des situations "d'enchevêtrement d'alliances, où coexistent entre les mêmes entreprises des rapports de coopération et des rapports de compétition, et où on ne sait plus clairement qui est allié et qui

est concurrent" (Gérard Dréan, 1996 A, p.4 et p. 326). C'est ce que confirme une étude consacrée à l'industrie québécoise du logiciel, qui montre que si les entreprises qui réalisent des alliances ont de meilleures performances, le succès d'un accord est lié à la poursuite d'un objectif unique, soit commercial, soit technologique (Jorge Niosi, 2000).

On peut donc considérer que *les phénomènes de diversification des entreprises ne remettent pas en cause la tendance dominante à la spécialisation et à la désintégration de l'industrie informatique*. En effet, dans le cas de rachat d'entreprise ayant une activité distincte, il est de plus en plus fréquent que celui-ci s'effectue en laissant une autonomie significative à l'entreprise rachetée, comme le montre l'exemple d'IBM qui s'est bien gardé d'intégrer juridiquement Lotus lors de l'OPA sur cette entreprise. Dans certains cas, à l'inverse, des activités qui avaient été intégrées se voient dotées d'une autonomie plus importante : par exemple, la société Network Associates (acteur dominant dans les logiciels de sécurité informatique et dixième entreprise mondiale de logiciels) qui avait intégré l'activité de ses différentes acquisitions (pour un montant de 2,5 milliards de dollars) les fractionne en quatre unités distinctes et autonomes, devant disposer à terme de leur propre capitalisation boursière : les logiciels antivirus pour McAfee Inc., les logiciels de cryptage et de réseaux privés virtuels pour PGP Security Inc., les logiciels de service de dépannage pour Magic Solutions Inc. et les outils de gestion réseau pour Sniffer Technologies Inc. (Les Chroniques de Cybérie, 18 janvier 2000). La multiplication des relations diverses de coopération entre entreprises - phénomène réel même s'il est moins important que ce que pourrait laisser supposer un simple dénombrement des annonces d'accords conclus – traduit moins un affaiblissement de la spécialisation des entreprises que le développement de réseaux. Ces réseaux d'alliances entre firmes indépendantes, formes intermédiaires entre le marché et la hiérarchie, où se mêlent étroitement des rapports de coopération et de compétition (*coopétition*) sur le modèle des relations entre pionniers de l'informatique dans la Silicon Valley à ses débuts, présentent la particularité de pouvoir "cumuler les avantages d'expertise technique propre à la structure fonctionnelle, de capacité de réponse au marché de la structure divisionnelle et l'équilibre des préoccupations de la structure matricielle" (Alain Desrumeaux, 1997, p. 3160).

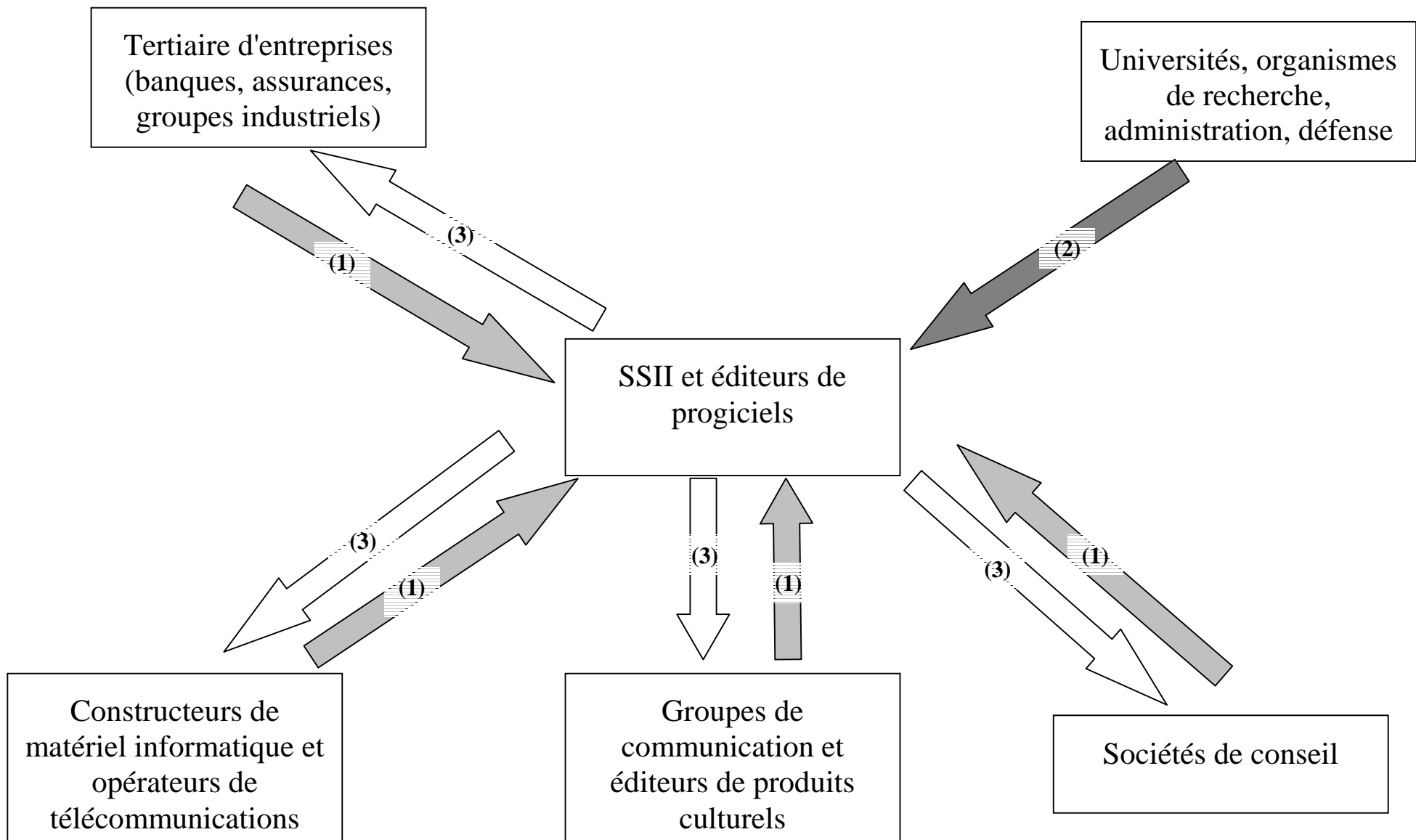
En conclusion on peut représenter la diversité des producteurs de logiciels et ses évolutions principales de la façon suivante (cf. schéma XI page suivante).

Les trajectoires (1) qui aboutissent à une augmentation du poids des entreprises dont la production de logiciels est l'activité principale (SSII et éditeurs de progiciels) recouvrent deux phénomènes : le premier, historiquement parlant, correspond au mouvement d'externalisation et de filialisation des activités logicielles de la part d'entreprises dont l'activité principale était autre (principalement les constructeurs informatiques et les grands groupes industriels et de services). Le deuxième phénomène, plus récent et plus limité, est constitué par la diversification de SSII et d'éditeurs de progiciels puissants, dans des activités complémentaires de leur production principale, notamment par des rachats et des prises de participation (principalement vers le conseil, les services et les produits culturels multimédias).

La trajectoire (2) correspond à la création de sociétés par la valorisation de logiciels et de compétences développés initialement dans un cadre public (essentiellement les universités et les organismes de recherche).

Les trajectoires (3) correspondent à l'attrait, stratégique et financier, de la production de logiciels pour des entreprises diverses, extérieures à cette activité. On peut noter que cette évolution est moins importante que l'évolution de type (1) et qu'elle n'est pas symétrique : dans la plupart des cas, les prises de participation et rachats de SSII et d'éditeurs de progiciels se limitent à l'existence de liaisons financières qui n'impliquent pas la ré-internalisation de la production des logiciels.

Schéma XI : Les différents producteurs de logiciels



La production des logiciels a été et est réalisée par des acteurs très divers, phénomène qui dépasse la simple existence de "groupes stratégiques" distincts (Michael E. Porter, 1982)⁵⁴. En prenant en compte la diversité des produits et des utilisateurs examinée dans la section I, l'économie du logiciel apparaît comme caractérisée par une diversification importante. Un pas supplémentaire dans l'analyse consisterait à intégrer dans un même cadre théorique la diversité croissante de l'économie du logiciel et les questions critiques qu'elle doit affronter et qui ont été traitées dans le chapitre précédent. C'est ce que nous nous proposons de faire en mettant en évidence l'existence de quatre "mondes de production" des logiciels.

⁵⁴ La notion de groupe stratégique désigne l'existence d'un ensemble de firmes qui utilisent le même arsenal de mesures stratégiques pour se concurrencer, et vise à expliquer la coexistence durable, vérifiée empiriquement dans certains secteurs, d'entreprises aux caractéristiques très différentes.

Chapitre VII

LES QUATRE "MONDES DE PRODUCTION"

DE L'ECONOMIE DU LOGICIEL

Nous avons vu que l'économie du logiciel se caractérise par une extrême diversité des produits, des producteurs et des utilisateurs, et par l'existence permanente de questions critiques (fiabilité, adaptation aux utilisateurs et productivité) qui reçoivent des réponses différenciées. Pour approfondir cette analyse, il est nécessaire de s'appuyer sur un cadre théorique permettant d'interpréter l'existence de formes de coordination multiples. De ce point de vue, la théorie néoclassique semble inadaptée. En effet, les deux postulats principaux de l'axiomatique néoclassique sont que les agents économiques se comportent de façon à maximiser leur utilité (rationalité maximisatrice), et que le marché est la meilleure forme de coordination, celle qui conduit à l'efficacité optimale au sens de Pareto. Certes, la théorie néoclassique ne se limite pas à cette axiomatique : la notion de maximisation a été assouplie par l'élargissement de la notion d'"utilité", et d'autres modes de coordination que le marché ont été intégrés. Malgré tout, la norme de référence du comportement des individus reste la seule recherche d'un gain maximal, et le marché est considéré comme la forme "normale" de coordination. A l'inverse, l'étude de l'économie du logiciel montre l'importance de mécanismes de coordination non marchands (règles, conventions, institutions) - d'où la diversité des marchés et le fait qu'une part non négligeable de la production des logiciels est réalisée sans recourir à un quelconque marché -, et l'existence d'une pluralité de registres d'action, qui ne se limitent pas à recourir au calcul économique privé.

C'est à partir d'un cadre théorique alternatif, intégrant au centre de sa construction les dimensions sociales et historiques (épaisseur temporelle des problèmes, irréversibilité et "cumulativité" des choix ...), qu'il nous semble possible de penser la diversité des attentes et des besoins des utilisateurs, l'hétérogénéité des producteurs, et la pluralité des formes de coordination entre les agents. Ce cadre - que l'on peut qualifier de socio-économique - est utilisé pour construire une typologie de l'économie du logiciel en quatre mondes de production. Cette typologie permettra d'analyser comment les différents mondes de

production du logiciel se positionnent différemment face aux trois dimensions critiques des logiciels, à savoir la productivité dans la production des logiciels, la fiabilité des logiciels produits, et l'adéquation aux besoins des utilisateurs. La typologie adoptée est reprise, avec quelques modifications, de celle que Robert Salais et Michael Storper (1993) ont élaboré, pour appréhender, au niveau macro-économique, la pluralité des modèles de systèmes productifs qui peuvent coexister au sein d'une économie nationale. L'originalité de cette typologie est qu'elle est centrée sur *l'identité du produit*, analysée comme le produit des conventions existantes entre l'ensemble des acteurs économiques concernés, producteurs mais aussi utilisateurs. Son élaboration, comme son exposition, nécessitent de réexaminer certaines des questions clés de la science économique, concernant la nature de la rationalité économique, l'analyse des multiples incertitudes qui affectent la vie économique, le rôle des institutions et des conventions dans la coordination entre les agents économiques.

C'est pourquoi nous consacrerons la première section à un examen de la théorie des mondes de production, avant de voir dans une deuxième section l'application de cette théorie à l'économie du logiciel.

Section I - La théorie des "mondes de production" revisitée

A - UN CADRE THEORIQUE SOCIO-ECONOMIQUE

Le point de départ est l'analyse de l'incertitude et plus particulièrement de l'incertitude radicale (1). Ceci contribue à enrichir la conception de la rationalité des agents économiques en montrant l'importance des formes "non instrumentales" de la rationalité, d'où une approche de nature davantage socio-économique plutôt que strictement analytique (2). Cette approche permet une analyse des institutions plus riche que les explications en termes de défaillance du marché (3), et souligne toute l'importance des conventions (4), qui sont à la base de l'existence de plusieurs "mondes de production".

1 - Incertitudes

"Ainsi placés en situation d'avoir à coordonner leurs actions de travail, d'offre et de demande (sinon le produit ne pourrait arriver à réalité), les acteurs économiques doivent être considérés comme s'affrontant à une *incertitude radicale* : sur le futur, sur l'action de l'autre,

sur ses projets et ses attentes, sur les usages des objets présents" (Robert Salais et Michael Storper, 1993, p.13). Les incertitudes auxquelles sont confrontés les agents économiques (individus ou organisations) sont multiples : les incertitudes sur les comportements des autres agents (en raison notamment de la possibilité d'opportunisme) ont fait l'objet de multiples travaux à travers l'analyse des phénomènes d'aléa moral et de sélection adverse. Pourtant il est fréquent que l'incertitude principale porte davantage sur l'"objet" de l'échange, bien ou service (Jean Gadrey, 1996 A, p. 37-38). C'est notamment le cas pour de multiples activités de services où le produit est "une construction sociale, incertaine, conflictuelle, multidimensionnelle" (idem, p. 293). La production d'un logiciel sur mesure, qui peut être considérée comme une activité de services, comporte des incertitudes majeures sur les résultats de la prestation, ce dont témoigne la fréquence des recours aux tribunaux. Dans la création d'un logiciel sur mesure, le destinataire du service est fortement impliqué dans le processus d'obtention du résultat (Serge Bouchy, 1994, p. 314) ce qui augmente encore l'incertitude en raison de l'impossibilité de normes comportementales prévisibles des clients : comme le note Jean Gadrey, "on ne recrute pas ses clients comme ses salariés" (1996 A, p. 287).

Une autre source d'incertitude réside dans la technologie. En général, les incertitudes sur la technique sont jugées beaucoup moins importantes que les incertitudes sur l'état du marché (Michel Callon, 1993, p. 12). C'est certainement exact quand la technologie n'est considérée que sous l'angle de son influence sur les *inputs* de l'activité. Ainsi Claire Charbit et Jean-Benoît Zimmermann définissent "l'incertitude technologique à travers trois problèmes emboîtés de nature informationnelle et décisionnelle (...) : quelles sont les technologies disponibles ? Quelle est la technologie la plus appropriée aux besoins de la firme ? Comment utiliser au mieux la technologie retenue ?" (1997, p. 10). Mais dans le cas de la production des logiciels, l'incertitude technologique concerne également *l'output* de l'activité, à savoir un logiciel qui est lui-même une technologie, qui doit impérativement s'intégrer à d'autres technologies (logicielles, matérielles, voire organisationnelles), dont on a souligné la rapidité d'évolution et leur caractère difficilement prévisible. Pour ce type de production, il n'est pas possible de séparer incertitude technologique et incertitude sur l'état du marché. Dans certains cas, l'existence (ou non) d'un marché pour le produit envisagé peut être complètement déterminée par la nature de l'ensemble des technologies complémentaires qui s'imposent *in fine*. Cette incertitude technologique existe dans la production des logiciels sur mesure : par exemple, les sociétés qui ont négligé de développer des compétences dans les architectures

client-serveur se retrouvent dans des positions difficiles dans des domaines d'activité de plus en plus divers. Mais cette incertitude technologique a un caractère encore plus décisif dans la production des progiciels, vu le caractère difficilement réversible des décisions, en raison de l'importance des coûts fixes, largement irrécupérables, préalables à la commercialisation d'un progiciel : une société qui aurait axé sa stratégie de développement de progiciels sur l'avènement des ordinateurs de réseaux ou NC (cf. chapitre II) n'aurait quasiment réalisé aucune vente ; inversement, il était impératif pour les sociétés d'intégrer à leurs produits des fonctionnalités liées à Internet pour simplement survivre sur le marché. Pourtant, dans les deux cas (échec du NC, formidable succès du développement d'Internet), les prévisions sont extrêmement délicates en raison des mécanismes de développement cumulatifs de ces technologies, déjà analysés : rôle déterminant des effets de feed-back positifs et négatifs, importance des petits événements initiaux...

De façon générale, la période actuelle se caractérise par un niveau "intrinsèque" d'incertitude notablement plus élevé que par le passé (Godefroy Dang Nguyen, Pascal Petit, Denis Phan, 1997, p. 70)¹. Les justifications avancées par ces auteurs s'appliquent particulièrement bien à l'économie du logiciel : les sociétés évoluent vers plus de complexité et les signaux qu'elles utilisent demandent de grandes capacités de lecture et d'interprétation qui multiplient les risques d'erreur ; elles développent des moyens d'action et de réaction particulièrement efficaces, ce qui accroît en cas d'erreur les catastrophes. Paradoxalement, la croissance de l'information et le progrès des connaissances peuvent augmenter le niveau général d'incertitude, notamment en nous permettant d'évaluer l'étendue de notre ignorance : par exemple, les théories du chaos (effet "aile de papillon"), en montrant la force des interdépendances, nous font percevoir la "présence d'un incertain radical, non réductible à une probabilisation d'événements qui faciliterait sa représentation et son intégration dans les schémas rationnels" (Pascal Petit, 1998, p. 33 et 386).

La distinction entre un risque probabilisable et une incertitude véritable avait été formulée par Frank H. Knight dès 1921² : "la différence pratique entre les deux catégories, le

¹ Cette augmentation de l'incertitude est certainement un des facteurs déterminants du développement important des activités de conseil.

² Bien évidemment l'insistance sur cette distinction ne signifie pas l'absence d'un continuum de situations entre ces deux cas polaires (cas similaires / cas uniques), ce que reconnaissait Frank H. Knight, qui soulignait par ailleurs que l'incertitude n'est pas que négative : "dans chaque cas, nous nous efforçons effectivement de réduire l'incertitude, bien que nous n'aimerions pas qu'elle soit éliminée de notre vie" (1964, p. 238). D'une certaine

risque et l'incertitude, est que s'agissant de la première, la distribution du résultat parmi un ensemble de cas, est connue (soit par le calcul *a priori*, soit par des statistiques fondées sur les fréquences observées), tandis que ceci n'est pas vrai de l'incertitude en raison de l'impossibilité de regrouper les cas, la situation à traiter présentant un degré élevé de singularité" (1964, p. 233). Or l'incertitude véritable, qui peut concerner aussi bien le déroulement du processus de production que l'évolution de la demande, "détruit la possibilité d'agir à partir d'une connaissance scientifique, rationnelle du monde" (Jean-Daniel Reynaud, 1989, p. 170). Quand il n'est pas possible d'opérer des regroupements dans des classes homogènes pour permettre le traitement statistique de l'incertitude et sa transformation en un risque probabilisable, l'action ne peut s'effectuer qu'en fonction d'estimations (au sens d'opinions, de convictions, de jugements) et non d'une probabilité subjective tendant vers une probabilité objective (Dorothee Rivaud-Danset, 1998, p. 30). On peut de plus ajouter que, pour certains auteurs, le caractère radical de l'incertitude ne se limite pas à des explications cognitivistes (asymétries d'informations, caractère non probabilisable des événements futurs) mais provient du caractère conflictuel des rapports sociaux : par exemple, Thomas Coutrot (1998, p. 153) souligne l'existence d'une "incertitude marchande" générée par les rapports marchands (concurrence entre les producteurs), d'une "incertitude organisationnelle" et d'une "incertitude sociale" générées par les rapports capitalistes (antagonisme entre employeurs et salariés).

La présence d'une incertitude radicale implique que l'action économique ne pourrait se dérouler si elle résultait uniquement de calculs maximisateurs privés, comme le postule le modèle de la rationalité de la théorie standard.

2 - Rationalités

La notion de rationalité n'est pas simple à définir si l'on entend dépasser la définition des dictionnaires comme étant la capacité à exercer sa raison. Les multiples controverses sur la vision plus ou moins large de la notion de rationalité recouvrent les débats sur la nature des motivations humaines, ici dans le domaine économique. Jean-Daniel Reynaud dit d'une action qu'elle est rationnelle s'il "est possible à un observateur qui prend la peine de s'informer d'en comprendre les motivations et les objectifs" (1989, p. 278). Il précise que le terme rationalité

manière, la nouveauté, l'invention sont contradictoires avec l'existence de situations qui se limiteraient à être des "risques probabilisables".

a "un sens très faible : l'acteur a une intention et un projet et l'un et l'autre ne sont pas absurdes aux yeux de l'observateur" (idem, p. 238). De même, Jürgen Habermas (1987) définit une action rationnelle comme étant compréhensible par d'autres : l'action de l'individu A sera rationnelle si elle est susceptible d'être critiquée, approuvée ou réfutée par un individu B pour de bonnes raisons, c'est à dire pour des raisons qui seraient aussi jugées bonnes pour d'autres individus à la place de B. Par contre, pour de nombreux auteurs, la notion de rationalité est beaucoup plus limitative que la capacité à trouver une raison à une action : elle correspond à la capacité à choisir en comparant, par anticipation, les choix entre différentes options et fait intervenir à titre central la notion de calcul en fonction de l'intérêt personnel.

La rationalité substantielle

Cette conception étroite de la rationalité est celle de la théorie économique standard. La rationalité de l'*homo economicus* néoclassique, qualifiée de rationalité omnisciente ou substantielle³, présente les caractéristiques suivantes : c'est une rationalité de type *instrumental* (ou rationalité en finalité au sens de Max Weber) où le seul critère est l'efficacité des moyens choisis par rapport à un but donné, une rationalité de type *calculatoire* où l'agent économique peut comparer les avantages et les coûts des différentes actions possibles, et une rationalité de type *optimisatrice* où l'action rationnellement choisie correspond à l'optimum par la maximisation de l'utilisation des ressources limitées disponibles. Selon cette conception, parfois qualifiée de conséquentialiste (on juge les actes uniquement en fonction des conséquences), toutes les actions s'expliquent uniquement par la recherche de l'intérêt personnel⁴ : l'homme est réduit à un animal préoccupé par ses seuls intérêts, un "égoïste intéressé" (Amartya Sen, 1993, p. 93). Cet auteur souligne que "l'optique de l'"égoïsme par définition" parfois dénommée choix rationnel, ne suppose rien d'autre que la cohérence interne " (idem, p. 94) dans le comportement de choix, et que le flou des notions d'"utilité" ou d'"intérêt personnel" rend le raisonnement souvent tautologique⁵ : en effet, si l'on définit,

³ L'expression "rationalité substantive" est parfois également employée - notamment par Herbert A. Simon (1978) - pour désigner le fait que son contenu est l'équivalent d'un calcul ou d'un algorithme et que la solution de ce calcul désigne la *substance* du choix à effectuer par l'individu.

⁴ Par exemple dans son ouvrage *Mathematical Psychics* (1881), Edgeworth affirme que "le premier principe de la science économique veut que tout agent ne soit mû que par son propre intérêt" (cité par Amartya Sen, 1993, p. 87).

⁵ "La stratégie la plus simple, citée souvent en exemple de l' "impérialisme" économiste le plus brutal, consiste à dévoiler, sous tous les comportements sociaux (...), des actes qui relèvent en fait d'un intérêt

comme c'est fréquemment le cas, l'utilité ou les préférences de l'individu à partir de ce que révèle son comportement, on peut considérer que quoique fasse un individu, il maximise son "utilité" et qu'il a donc un comportement rationnel. De même, Albert O. Hirschman note que le concept d'intérêt a été tellement "dilué" que "les intérêts en vinrent à parcourir toute la gamme des actions humaines, de l'égoïsme le plus étroit à l'altruisme le plus large" et que l'intérêt "se trouva interprété comme le moteur de tout ce que les hommes font ou désirent faire" (1986, p. 24).

La "futilité de toute tentative pour réduire l'action humaine à un motif unique, fût-ce l'intérêt" (Albert O. Hirschman, 1986, p. 27) est une des nombreuses critiques du modèle de l'action rationnelle de la théorie néoclassique. Mark Granovetter montre que "la poursuite d'objectifs économiques s'accompagne normalement de celle d'autres objectifs de nature non-économique, tels que la sociabilité, l'approbation, le statut social et le pouvoir" (1994, p. 81). Amartya Sen souligne que l'action peut se rapporter parfois à un sens de l'obligation qui dépasse les conséquences de l'action, et que certains actes n'entraînent aucun gain personnel, mais sont justifiés par le respect de certaines règles de comportement : "l'exclusion de toute considération autre que l'intérêt personnel semble imposer une limite totalement arbitraire à la notion de rationalité" (Amartya Sen, 1993, p. 114). Réfléchir sur les fins de l'action elles-mêmes ou se poser la question de la valeur intrinsèque des moyens mis en œuvre (indépendamment de leurs conséquences), c'est introduire des questions éthiques ou déontologiques⁶ qui sont hors-sujet dans le modèle standard.

Une autre critique concerne le postulat des préférences de l'acteur qui sont considérées comme données une fois pour toutes, cohérentes et hiérarchisées : "on attribue à la personne *un seul* classement des préférences, et, au gré des besoins, ce classement est supposé refléter les intérêts de la personne, représenter son bien-être, résumer son opinion sur ce qu'il convient de faire, et décrire ses choix et son comportement effectifs" (Amartya Sen, 1993, p. 106). Cet auteur considère que si "une personne ainsi décrite peut être "rationnelle" au sens limité où elle ne fait preuve d'aucune incohérence dans son comportement de choix", elle est "à vrai dire un demeuré social, (...), un idiot rationnel drapé dans la gloire de son classement de

individuel bien compris (...). Les justifications des acteurs sont tenues pour des illusions trompeuses quant à la nature véritable de leur acte" (Pierre Livet et Laurent Thévenot, 1994, p. 141).

⁶ Pour Amitai Etzioni (1988), la déontologie est le fait de porter un jugement sur des actes ou des intentions du point de vue de leur conformité à des principes, devoirs ou obligations, et non pas du point de vue de leurs conséquences.

préférences unique et multifonctionnel" (idem, p. 107). Nicolas Postel souligne que cette conception de l'individu, qui articule des compétences logiques et calculatoires très importantes, et des compétences pratiques nulles, élimine tout rapport aux autres et toute temporalité réelle (2000, p. 49). James G. March (1991) fait observer que les préférences individuelles sont souvent floues et incohérentes, changent avec le temps, en partie à la suite des actions entreprises, et que les objectifs des acteurs sont vagues, problématiques et instables. Plus généralement, cet auteur met en évidence les écarts des comportements de choix observés dans la réalité, par rapport à la ligne tracée par les modèles classiques. Il considère que ces écarts ne doivent pas être considérés d'un point de vue normatif comme des erreurs à corriger, et qu'au contraire, il faut partir de l'hypothèse que tout comportement réel en matière de choix est sensé d'une certaine façon : les comportements observés sont non seulement compréhensibles en tant que phénomènes humains mais dans une grande mesure intelligents (idem, p. 142) et en conséquence c'est le modèle de la rationalité qui doit être modifié.

La rationalité limitée et la rationalité procédurale

Un des exemples les plus célèbres d'une telle modification est le modèle de la rationalité limitée⁷ de Herbert A. Simon. Au départ, il ne s'agit que de petites modifications de la théorie du comportement économique (James G. March, 1991, p. 139) intégrant de nouvelles contraintes qui conduisent à remettre en cause le postulat selon lequel l'acteur a toutes les informations et une capacité illimitée pour leur traitement. Herbert A. Simon souligne que les ressources cognitives des individus sont limitées, et qu'ils n'ont pas une capacité infinie de recherche, de traitement et de stockage de l'information. Pour Herbert A. Simon, lorsque l'environnement et le problème à résoudre sont simples, la théorie de la rationalité standard peut s'appliquer, ce qui n'est plus le cas lorsqu'ils deviennent complexes : "dans des situations complexes, il y a fréquemment un écart important entre l'environnement réel de la décision (tel que le percevrait Dieu ou tout autre observateur omniscient) et l'environnement tel que les acteurs le perçoivent" (1978, p. 5). Interviennent les informations dont dispose l'acteur qui dépendent de sa position pour que celles-ci lui parviennent et de l'habileté avec laquelle il a su trouver les plus pertinentes. Quant au problème à résoudre, sa

⁷ C'est ainsi qu'a été traduit "bounded rationality". Nicolas Postel (2000, p. 326) considère que "rationalité bornée" exprimerait mieux l'idée selon laquelle l'individu n'est rationnel que dans la limite de ses connaissances et de ses capacités cognitives.

complexité est envisagée dans son sens informatique⁸ : son algorithme de résolution peut être simple (exemple classique de la minimisation du trajet du voyageur de commerce), mais la croissance des ressources nécessaires à son exécution augmente de façon exponentielle avec la taille du problème (cf. chapitre III). La rationalité n'est donc plus fondée sur les caractéristiques "objectives" de la situation, mais elle est "subjective", relative au niveau de connaissance de l'individu et à ses compétences pour traiter les informations. Si la prise en compte du coût et des caractéristiques de la collecte et du traitement de l'information permet d'enrichir la représentation du comportement des individus, la rationalité limitée peut être considérée comme étant encore de type calculatoire.

Par contre ce n'est plus le cas pour le concept de "rationalité procédurale" introduit par Herbert A. Simon à partir de la seconde moitié des années soixante dix. La rationalité procédurale n'est pas une autre forme de rationalité maximisatrice limitée par l'existence d'un environnement complexe et par les "très modestes moyens de computation" dont disposent les individus, mais une autre forme de rationalité qui rompt avec le postulat de la recherche de la solution optimale parmi toutes les options possibles. Avec la rationalité procédurale, Herbert A. Simon oppose à l'attitude optimisatrice, le principe du seuil de satisfaction de l'individu (*satisficing*) : les agents mettent en œuvre des stratégies pour parvenir à des situations leur paraissant préférables dans un processus d'ajustement mutuel du possible et du désirable. Dès lors, la rationalité concerne moins le résultat de la délibération (la décision), que les mécanismes de recension et d'interprétation de l'information pour "comprendre" la situation et prendre une décision. Selon Herbert A. Simon, les procédés cognitifs par lesquels l'individu rationnel parvient à se donner une représentation adéquate d'un environnement complexe qu'il ne peut de toute façon pas connaître parfaitement, sont en grande partie pratiques et intuitifs, et passent par l'activation de routines et d'heuristiques en repérant des analogies entre des situations récurrentes. Cette vision du comportement humain s'apparente à celle de Keynes, à la différence que Simon s'intéresse également à l'analyse des procédés cognitifs effectivement mis en œuvre par les individus pour ramener une situation complexe à une alternative décidable, ce qui ouvre les sciences économiques aux sciences cognitives et à la psychologie.

⁸ Rappelons que parmi les multiples disciplines étudiées par Herbert A. Simon figure en bonne place l'informatique. Sa vision de la rationalité d'un individu comme devant intégrer une gestion rationnelle de ses ressources cognitives, est étroitement liée à ses recherches sur les ressources informatiques nécessaires pour résoudre des problèmes d'intelligence artificielle.

Cependant, les conceptions de la rationalité de Herbert A. Simon, qu'il s'agisse de la rationalité limitée ou de la rationalité procédurale, restent des rationalités de type instrumental, caractérisées par la recherche de la meilleure efficacité possible dans l'utilisation des moyens par rapport à une fin considérée comme donnée. Il s'agit d'une conception essentiellement "technique" de l'efficacité qui se situe dans un espace où une cause produit nécessairement un effet prévisible, et où la question principale est celle de la bonne adéquation des moyens aux fins, sans porter de jugement ni sur l'objectif visé, ni sur les moyens utilisés. Or, pour certains auteurs, la rationalité ne peut se limiter à ce cadre.

Des formes de rationalité non instrumentales

Ainsi, Feriel Kandil estime que la rationalité "concerne également une réflexion sur les fins poursuivies dans l'action" (1998, p. 147) et propose de "rendre compte des actes économiques en termes de raison pratique plutôt que de rationalité instrumentale [ce qui] implique de porter son attention sur la multiplicité des capacités de jugement en exercice dans la délibération, elle-même prise dans le temps de réalisation de l'action" (idem, p. 166). Elle nomme "l'intelligibilité pratique de l'action", "le fait que l'action effective ait un sens qui, par sa singularité, échappe à toute détermination générale en termes de calcul de bien-être" ; elle précise que "ce sens est réfléchi dans le cours d'interaction à partir des capacités de jugement de l'acteur" et que "ces dernières s'appuient sur les différentes formes du collectif présentes dans la situation d'action" (idem, p. 150).

Amartya Sen insiste sur l'existence "de *l'engagement* comme composante du comportement" (1993, p. 116). A la différence d'un comportement fondé sur la *compassion* qui reste "en un sens important, égoïste", car "le souci d'autrui influe directement sur votre propre bien-être" (idem, p. 97), l'engagement "établit une distance entre choix personnel et bien-être personnel" (idem, p. 100) : en effet, dans un comportement basé sur l'engagement, "une personne choisit une action qui, pense-t-elle, lui apportera un degré de bien-être personnel inférieur à celui que lui procurerait une autre action qu'elle pourrait aussi mener" (idem, p. 98). Autrement dit, elle choisit, en connaissance de cause, de ne pas chercher son intérêt au nom d'une valeur qui s'impose à elle. Si dans certains cas, un individu peut adopter un comportement altruiste contraire à ses préférences, c'est qu'il possède en fait plusieurs systèmes de préférences, le choix du système de préférence sur la base duquel il va effectuer

ses choix effectifs, étant un méta-choix, de type éthique le plus souvent⁹, renvoyant à l'existence d'un "méta-classement" ou "classements de classements de préférences" (idem, p. 108).

Albert O. Hirschman reprend la distinction de Sen entre les préférences et les métapréférences (1986, p. 90), mais en soulignant l'existence d'un "lien étroit entre le changement de préférence et le concept de métapréférence", il introduit une distinction importante entre les "diverses sortes de modifications de préférence" (p. 92) : d'une part, des "modifications qui surviennent sans que l'on observe toute cette complexe formation antérieure d'une métapréférence" qui ne sont que "des changements de goûts impulsifs, spontanés, fortuits, passagers, induits par la publicité et en général peu importants" mais qui sont les modifications "qui ont été le principal objet de l'analyse économique dans la théorie de la consommation" ; d'autre part, "des changements délibérés et difficiles, ceux qui sont précédés par la formation d'une métapréférence qui entre en conflit avec la préférence observée et pratiquée jusque là" (idem). Dans la mesure où cette préférence prête à discussion (que ce soit avec d'autres ou avec soi-même), il ne s'agit plus d'un simple goût mais d'une *valeur*. Dans ce cas, l'attitude habituelle des économistes néoclassiques, qui considèrent les changements de préférences comme "dénusés d'intérêt analytique"¹⁰, n'est pas acceptable : un changement de valeurs peut être "autonome et délibéré" (idem, p. 95), les gens étant "capables de modifier leurs valeurs" (idem, p. 93) ; en conséquence, les évolutions des valeurs et leurs effets sur les comportements méritent d'être analysés¹¹.

Il est toutefois évident que les choix de valeurs ne peuvent être étudiés valablement selon le modèle du calcul des moyens les plus efficaces pour accaparer le maximum de biens, ce qui conduit à une deuxième idée importante d'Albert O. Hirschman, à savoir "qu'il devient essentiel de s'apercevoir que les modèles fins/moyens et coûts/bénéfices sont loin de recouvrir tous les aspects de l'expérience et de l'activité humaines" (1986, p. 100). Une preuve en est

⁹ "Un classement des préférences peut être à coup sûr plus éthique qu'un autre et moins qu'un troisième" (Amartya Sen, 1993, p. 108).

¹⁰ Albert O. Hirschman cite un article de Gary Becker et George Stigler (1977) significativement intitulé "De Gustibus Non Est Disputandum" (*American Economic Review*, vol. 67, pp. 76-90), où tout changement de préférence est assimilé à de simples altérations des "goûts insondables et souvent capricieux".

¹¹ Une tentative en ce sens est celle de James G. March (1991) qui analyse comment nous construisons et gérons nos préférences.

"l'existence et l'importance des activités non utilitaires (...) dont l'issue reste si incertaine [et qui] sont étrangement caractérisées par une certaine fusion (et confusion) entre la recherche et le but" (idem, p. 98-99). Albert O. Hirschman constate que "très tôt, hommes et femmes ont consacré une bonne part de leur temps à des entreprises dont le succès est tout simplement imprévisible. Il s'agit par exemple, de la recherche de la vérité, de la beauté, de la justice, de la liberté, de la communauté, de l'amitié, de l'amour, du salut, sans parler d'aspirations moins édifiantes comme la gloire ou la vengeance" (idem, p. 97). Pour ces activités, où souvent "le but est lointain et l'aboutissement tout à fait problématique", la "fusion de la recherche et du but" réside dans la possibilité de pouvoir "savourer à l'avance" certains objectifs de l'action : "qui cherche la vérité (ou la beauté) acquiert souvent la conviction, si fugitive soit-elle, de l'avoir atteinte ou touchée du doigt ; qui participe à un mouvement pour la liberté et la justice a fréquemment l'impression d'en profiter déjà" (idem, p. 99). Albert O. Hirschman ajoute que "l'action non utilitaire en général contribue à créer ou à renforcer le sentiment d'être une véritable personne" et "peut donc être considérée par l'économiste comme un investissement non dans le capital humain mais dans quelque chose de beaucoup plus fondamental : l'identité individuelle ou de groupe" (idem, p. 99-100).

Ce faisant Albert O. Hirschman franchit ce que Robert Salais appelle "la grande ligne de partage, celle qui sépare ceux qui en restent à la rationalité individuelle opportuniste (...) de ceux qui tiennent pour acquis la présence dans l'action individuelle d'une référence générale" (1998 B, p. 265). Pour Robert Salais, "le problème n'est pas d'étendre, à nature inchangée, le calcul individuel rationnel individuel à la prise en compte de valeurs (comme l'altruisme) ou de l'intérêt des autres dans un jeu intertemporel de négociation anticipée, ni de se rabattre sur une rationalité qui, limitée par la complexité des circonstances, s'en tient à un choix de procédure satisfaisante" (idem, p. 284), mais de prendre en compte le fait que "les personnes mettent en œuvre une rationalité réaliste qui s'approche d'une forme de raison, une "raisonnabilité". Elles sont capables de dégager les conséquences de l'adoption d'une pratique au lieu d'une autre, d'adhérer au cours d'une action une fois celle-ci décidée, de résister aux tentations et attraits d'un gain immédiat" (idem, p. 265-266). Le calcul rationnel "demeure un des moments décisifs de l'action (et nombre de situations s'y laissent réduire)", mais "il faut l'intégrer dans une considération conjointe de tous les moments de l'action : délibération, décision et réalisation dans un cours singulier d'action collective" (idem, p. 284). Or, "une action individuelle ne peut espérer atteindre son but dans le cours d'une coordination que si elle manifeste qu'elle se réfère à "une bonne raison", soit à une raison d'ordre général (et non

individuelle ou sectorielle) –un principe de justice – qui peut être certes contestée, mais par rapport à une autre bonne raison et non en référence à un intérêt privé" (idem, p. 269). *Mettre en évidence la pluralité des logiques d'action, insister sur les impératifs de justification qui s'imposent aux acteurs dans la conduite de leur action sont au cœur du projet conventionnaliste : "replacer la rationalité intéressée au sein d'une pluralité de justifications possibles de l'action"¹².*

De même, Thomas Coutrot montre les difficultés rencontrées pour intégrer les dimensions collectives (culturelles ou sociales) des comportements, en gardant l'hypothèse de rationalité instrumentale individuelle. Il ironise en particulier sur "les tentatives théoriques de fonder l'émergence de la coopération sur des comportements non coopératifs", irrémédiablement "vouées à l'échec", car "la spécificité de la coopération est justement ... d'échapper à la rationalité instrumentale individuelle (...) : un comportement est coopératif s'il est soutenu par un autre type de rationalité, visant non la satisfaction d'un intérêt matériel personnel, mais la production ou la reproduction de ressources symboliques, par l'adhésion à des normes collectives" (1998, p. 91-92).

Une des analyses les plus stimulantes, qui différencie et intègre différents types d'actions rationnelles, est celle de Jürgen Habermas (1987). Selon cet auteur, au cours de l'histoire humaine se sont dégagés progressivement trois types d'actions rationnelles, en même temps que se différenciaient trois mondes : le monde objectif (l'ensemble des "entités sur lesquelles des énoncés vrais sont possibles"), le monde social ("l'ensemble des relations interpersonnelles reconnues comme légitimes par les gens concernés") et le monde de la subjectivité ("l'ensemble des événements vécus d'"accès privilégié, susceptibles d'expression véridique"). Le premier type d'action rationnelle, correspondant au monde objectif, est *l'agir téléologique ou instrumental*. Le critère de jugement qui s'applique est celui du *vrai*, et il correspond à la rationalité de l'*homooeconomicus* standard dont l'image est "l'engrenage des calculs égoïstes d'utilité, où le degré de coopération et de conflit varie en fonction des situations d'intérêt données". Le deuxième type d'action rationnelle est *l'agir régulé par les normes* qui renvoie au monde objectif et au monde social. Il aboutit à "l'accord socialement intégrateur, (...) réglé et stabilisé par la tradition culturelle et la socialisation", le critère de

¹² Voir le numéro spécial de la *Revue Economique* de mars 1989 consacré à l'Economie des Conventions où les auteurs précisent préférer "le terme de justifiable ou de raisonnable au terme de rationnel qui s'accorde plus difficilement à cette multiplicité" (Jean-Pierre Dupuy et alii, 1989, p. 148).

jugement pour valider ou critiquer l'action étant celui du *juste*. Le troisième type d'action rationnelle est *l'agir dramaturgique*. Il suppose l'existence de deux mondes, le monde objectif et le monde subjectif, et débouche sur la "relation consensuelle entre un public et ceux qui se présentent devant lui". Son critère de jugement est celui de *l'authentique*. A partir de ces trois types d'action rationnelle, Habermas définit *l'agir communicationnel*, qui se présente comme la forme la plus élevée de l'action. L'agir communicationnel produit une entente, et le consensus obtenu par la communication est d'un ordre supérieur parce qu'il suppose que les acteurs se sont accordés pour reconnaître que leurs expressions ou leurs actes sont à la fois vrais, justes et sincères, ce qui leur donne une force d'évidence ("la force sans violence du discours argumentatif"). En effet, pour qu'une action puisse être qualifiée de communicationnelle, il faut que "les plans d'action des acteurs participants ne [soient] pas coordonnés par des calculs de succès égoïstes, mais par des actes d'intercompréhension" (1987, p. 295) ; un acte d'intercompréhension est "conçu comme faisant partie d'un procès coopératif d'interprétation visant la définition intersubjective des situations" : les acteurs doivent prendre en compte les valeurs et la subjectivité d'autrui afin de réaliser l'entente.

Finalement, ce survol rapide montre que "s'éloigner de l'hypothèse standard de l'action rationnelle" conduit à une vision "délibérée" de l'action, dans le cours de laquelle "est présent du collectif sous une forme ou sous une autre" (Robert Salais, 1998 B, p. 263). Dans ces conceptions, l'individu apparaît comme "une personne, c'est à dire comme un être humain singulier, situé dans un milieu social défini dans le temps et l'espace, doté de valeurs, inscrit dans une trajectoire biographique" (idem, p. 270). Toute action, même apparemment uniquement "économique", en étant "une action significativement orientée" (Philippe Steiner, 1999, p. 37), a nécessairement une dimension sociale qui peut être plus ou moins importante. Dès lors, l'analyse des comportements ne peut se limiter à un strict cadre économique et doit intégrer nécessairement une composante sociale.

L'encastrement de l'économie dans le social ("embeddedness")

C'est la thèse défendue par Mark Granovetter, qui tente de se "frayer un chemin entre les conceptions sous- et sur-socialisées, en analysant la façon dont le comportement est "encastré" dans des systèmes stables de relations sociales" (1994, p. 85). Les approches "sous-socialisées" sont celles de la logique marchande pure, dans lesquelles l'individu est atomisé et ne connaît des autres que ce que lui dicte son intérêt personnel : dans ces approches, ne sont pas pris en compte le contexte social et historique de l'individu, l'influence

des structures collectives (institutions, règles sociales, organisations...) sur les comportements, et le fait que l'individu doit être d'emblée pensé comme un individu socialisé dont la rationalité est située¹³. A l'inverse, dans les approches sur-socialisées, les agents en sont réduits à être de simples supports de structures sociales et d'institutions qui détermineraient mécaniquement leurs comportements¹⁴. La conception de l'encastrement apparaît comme une voie méthodologique médiane entre une théorie individualiste de l'action sociale et une théorie holiste fondée sur la toute puissance des institutions : certes le comportement des acteurs est orienté par le calcul économique d'utilité, mais l'échelle des valeurs – ou des utilités –, sur laquelle porte ce calcul, la sélection des données pertinentes, voire la façon même de calculer, diffèrent en fonction des réseaux sociaux dans lesquels l'acteur évolue. Le comportement rationnel individuel est inséré dans des réseaux sociaux d'appartenance, qui confèrent un caractère éminemment social aux ressources et aux contraintes de l'individu¹⁵. Le concept d'*embeddedness* avait été introduit par Karl Polanyi en 1944, mais comme une caractéristique exclusive des sociétés précapitalistes, où la circulation des biens était basée sur la redistribution et la réciprocité et où donc l'activité économique était encadrée dans les relations sociales. En effet, pour Polanyi, le passage à une circulation des biens reposant principalement sur l'échange marchand dans un système autorégulateur de marchés, constitue une "grande transformation" marquée par le désencastrement des relations économiques vis-à-vis des relations sociales. A l'inverse, Mark Granovetter va montrer que, y compris dans les sociétés contemporaines les plus soumises à l'ordre marchand, l'économie reste encadrée dans les relations sociales¹⁶. Mark Granovetter en déduit la nécessité d'une sociologie économique qui "repose sur deux propositions sociologiques fondamentales : premièrement, l'action est toujours socialement située et ne peut pas être expliquée en faisant seulement référence aux motifs individuels ; deuxièmement, les institutions sociales ne

¹³ La rationalité est située au sens où "elle suppose un certain capital cognitif commun (au groupe considéré), sous la forme d'une communauté d'expériences, qu'elles soient culturelles ou historiques" (Robert Boyer, André Orléan, 1991, p. 236).

¹⁴ Selon cette conception, "les gens suivent automatiquement et inconditionnellement les coutumes, les habitudes ou les normes" (Mark Granovetter, 1994, p. 84).

¹⁵ "Le lieu de l'explication passe ainsi de la sphère de l'individu isolé à un cadre de référence plus large et plus social" (Mark Granovetter, 1994, p. 88).

¹⁶ Cf. notamment la célèbre étude de Granovetter sur le caractère décisif des relations amicales et familiales dans le processus, en apparence purement économique, de la recherche d'emploi.

jaillissent pas automatiquement en prenant une forme incontournable, mais sont construites socialement" (1990, p. 95-96).

3 - Les institutions

Une institution peut être définie "comme un ensemble de règles socio-économiques mises en place dans des conditions historiques, sur lesquelles les individus ou les groupes d'individus n'ont guère de prise, pour l'essentiel, dans le court et le moyen terme" (Claude Ménard, 1990, p. 15). Précisons que si une institution peut se présenter empiriquement comme "un système cohérent de règles, associé à une entité collective clairement identifiable et généralement sanctionné par le droit" (Olivier Favereau), il nous semble, dans la tradition des économistes "institutionnalistes", que les institutions englobent également des réalités moins formalisées : ainsi selon Thorstein Veblen les institutions sont "des habitudes acquises dans un contexte socio-institutionnel donné, et en relation avec ce contexte. Il s'agit d'habitudes épurées, figées, consacrées et socialisées". Cependant, qu'elles soient constituées de règles formelles ou informelles, les institutions constituent "des contraintes sociales exercées sur le comportement des agents économiques" (Fériel Kandil, 1998, p. 147).

Les institutions ont été réintroduites de façon spectaculaire dans l'analyse économique par Oliver E. Williamson (1975). Dans cette conception, l'existence des institutions est posée du point de vue de leur fonctionnalité, comme des arrangements efficaces, alternatifs au marché et en réponse aux déficiences de celui-ci, qui reste le point de départ de la coordination économique. L'émergence des institutions, qui permettent de minimiser les coûts de transaction, dérivent des calculs de maximisation des gains individuels des agents économiques. Au contraire, nous partageons la conception selon laquelle on ne peut déduire la présence des institutions des échecs du marché, et réduire les institutions à une somme d'arrangements contractuels. Certes les institutions sont des construits sociaux et, par-là même, des produits d'une intervention humaine mais si "la création des institutions [est] intentionnelle, en dernier ressort leurs effets et conséquences sont souvent très largement inintentionnels" (Robert Boyer, 1993, p. 5).

A l'inverse de la conception strictement microanalytique de Williamson, où toute institution peut être réduite à un calcul économique, l'analyse des institutions, dans l'approche que nous privilégions, doit intégrer d'autres dimensions explicatives, historiques et sociales. Si "il est illusoire de rechercher une explication monocausale des institutions économiques" (Robert Boyer, 1993, p. 4), on peut néanmoins s'appuyer sur des travaux, conventionnalistes

et régulationnistes, dans l'analyse des institutions. Ainsi pour Robert Salais, une institution est "tendue vers une unification sans cesse remise à plus tard, mue par ses contradictions internes et externes, composée de théories sociales, de textes et de règles, d'usages en situation et d'une chronique d'interprétation" (1998 B, p. 281). Les institutions sont "le fruit contradictoire de compromis entre plusieurs références générales, lesquelles sont mobilisées dans les actions des personnes et dans les jugements de possibilité qu'elles mènent" (idem, p. 280). Elles sont "prises dans le mouvement des choses, des pratiques et des interprétations, dans leurs interdépendances". Elles "sont inséparables de leur histoire et des débats auxquels elles donnent lieu pour leur fabrication, leur critique, leur ajustement" (idem, p. 281). "Le cœur du processus de fabrication [d'une institution] est un débat sur les normes qu'elle doit transcrire et mettre en œuvre", débat qui "acquiert son autonomie sociale"(idem, p. 284-285). Benjamin Coriat souligne que les institutions sont "largement autogénérées par les conflits entre groupes et classes" (1994, p. 125). Si dans sa constitution, une institution peut être appréhendée comme "un résultat du conflit des agents sur les règles du jeu", dans son fonctionnement elle peut être considérée comme un cadre de référence et de stabilisation dans lequel "les agents interagissent" (Robert Boyer, Yves Saillard, 1995, p. 67). Une institution peut être un outil de stabilisation pour la coopération d'acteurs aux intérêts potentiellement divergents, en spécifiant la place des acteurs, en façonnant le type de rationalité, en instaurant des systèmes d'incitations et de contraintes, en permettant de résoudre les déséquilibres et les anticipations croisées, et en réduisant les incertitudes. Robert Salais insiste sur le fait que les institutions permettent de "lever la paralysie engendrée par l'incertitude" (1998 B, p. 280) en répondant à une "attente qui n'est strictement ni cognitive, ni psychologique", concernant "des problèmes conjoints d'engagement, d'efficacité et d'équité" : en l'absence d'un "principe général ou valeur commune", l'action collective, correspondant à "la coordination économique en situations d'incertitude" ne pourrait exister ; en effet, une "coordination d'actions individuelles (...) dans un monde qu'à la fois elles entendent modifier et qui se modifie de manière imprévisible à mesure que ces actions progressent dans leur cours" ne peut s'effectuer que si "chacun a la garantie, si un aléa survient, qu'il sera traité avec justice" (idem, p. 256). De même, Ferial Kandil souligne que les institutions "instituent" l'action économique : "elles renferment un ensemble de savoirs collectifs (plus ou moins accessibles à chaque acteur économique) et elles donnent forme à l'action, en ce sens qu'elles dirigent l'action en fonction des champs de faisabilité qu'elles ouvrent" (1998, p. 148) ; quand la fin de l'action ne peut pas "être déterminée clairement et distinctement", quand elle n'est pas "réductible à une mesure quantitative de bien-être", l'acteur "pris dans un processus infini de réflexion et

d'interprétation sur les fins et sur les motifs de son action" ne pourrait agir sans règles institutionnelles comme "guide de l'action", comme "direction de sens" et comme "repère de signification" (idem, p. 154).

Contrairement aux théories qui opposent marchés et institutions, les institutions conçues comme des systèmes de règles englobent les marchés. En effet, loin d'être un produit naturel ou spontané, un marché est un construit social : un marché est "institué au sens où il nécessite un ordre, des règles, des contrôles" (Pierre-Yves Gomez, 1994, p. 52)¹⁷. Caractériser le marché comme étant une institution permet de souligner la pluralité et les transformations des formes possibles de l'institution marchande, un marché concret étant une construction collective et permanente des agents économiques qui y participent. Considérer que l'organisation n'est pas la seule véritable institution, conduit à faire de l'institution un concept général, dont le marché et l'organisation sont deux catégories particulières (Nicolas Postel, 2000, p. 113). Le marché peut être défini comme une institution fondée sur l'échange de ressources, où certes prédominent les accords marchands, mais qui suppose la présence d'autres règles centrées sur l'organisation des échanges. Une organisation est un ensemble formalisé aux frontières identifiables, qui se singularise par une fonction - l'usage de ressources – et par le fait qu'y dominant justement des règles d'usage et non pas des règles d'échange. Une entreprise est une organisation particulière, qui utilise des ressources pour produire d'autres ressources (Christophe Ramaux, 1996, p. 328-342). Toutes les institutions ont en commun d'être des ensembles de règles formant systèmes dans la mesure où ils sont dotés d'une certaine cohérence, les règles pouvant se décliner selon quatre modalités : le contrat, forme générique de l'accord explicite ; l'accord marchand, forme spécifique de l'accord explicite autour d'un couple prix-quantité ; la règle hétéronome, qui s'impose aux acteurs en dépit de leur volonté ; la convention, règle sur laquelle les acteurs s'accordent d'autant plus qu'elle n'apparaît pas comme le produit direct de leur volonté (idem, p. 324).

4 - Les conventions

Ce sont les conventions qui ont fait l'objet récemment d'un effort analytique particulier au point d'être la base d'un nouveau courant dénommé "l'économie des conventions". La thèse des économistes "conventionnalistes" est que "l'accord entre des individus, même lorsqu'il se

¹⁷ Cf. aussi Robert Salais et Michael Storper : "Le marché, même parfait, exige une organisation qui dépasse les agents" (1993, p. 48).

limite au contrat d'un échange marchand, n'est pas possible sans un cadre commun, sans une convention constitutive" (Jean-Pierre Dupuy et *alii*, 1989, p. 142). En réalité, l'économie des conventions s'est constituée autour de l'étude plus générale des différentes modalités de coordinations à l'œuvre dans le champ économique, en dépassant la seule existence des conventions¹⁸, avec trois directions privilégiées : la reconnaissance du caractère socialement construit des marchés dont l'étude nécessite de s'appuyer sur d'autres sciences sociales¹⁹, l'importance et l'étude de modes de coordination différents du mode marchand²⁰, et le fondement historique ou négocié des règles et institutions nécessaires à la réussite de toute coordination. Le constat effectué est qu'en raison des divergences d'objectifs des différents acteurs et de leurs connaissances incomplètes, une coordination optimale ne peut exister sans des "dispositifs interindividuels ou collectifs" qui assurent la "compatibilité des actions des agents économiques et (...) le respect des engagements" ; "ces dispositifs reposent sur la création de systèmes d'obligations mutuelles qui bornent les marges de manœuvre des agents et définissent des modalités d'interactions" (Eric Brousseau, 1997, p. 45).

La problématique adoptée s'écarte "à la fois de l'approche néoclassique pure du contrat (deux individus doués d'une même rationalité optimisatrice qui écrivent un accord complet prenant en compte toutes les occurrences possibles et tous les actes correspondants) et du postulat sociologique opposé selon lequel l'accord résulte d'une soumission des personnes à des normes exogènes qui s'imposeraient à eux" (Robert Salais, Michael Storper, 1993, p. 48). La démarche choisie tente de maintenir "une tension entre ces deux extrêmes, tension qui peut s'exprimer ainsi : (1) la coordination entre les agents économiques exige un cadre commun, une référence extérieure préalable ; (2) mais, en même temps, ce cadre commun est construit (ou ressaisi) dans le cours de leur action, par les agents eux-mêmes" (idem, p. 48). Dans la conciliation difficile entre ces deux présupposés quasi-contradictaires, les conventions

¹⁸ Olivier Favereau indique que le titre initialement envisagé pour le numéro spécial de la *Revue Economique* intitulé "L'économie des conventions", qui contient les textes fondateurs de ce courant, était "Règles et institutions". André Orléan, dans l'Introduction à *L'Analyse économique des conventions* (1994) déclare que "l'ambition du concept de convention" est de "comprendre comment se constitue une logique collective et quelles ressources elle doit mobiliser pour se stabiliser" (p. 16).

¹⁹ "Prendre en compte la réalité des relations marchandes, ce qui exige de traiter rigoureusement, en construisant un cadre d'analyse en partie original, la relation entre le modèle de marché et des notions qui lui sont à l'origine étrangères parce qu'elles trouvent leurs racines dans d'autres sciences de l'homme" (Dupuy et *alii*, 1989, p. 141).

²⁰ "Le rôle essentiel joué par les formes non marchandes de coordination, de production et d'allocation des ressources" (André Orléan, 1994, Introduction à *L'Analyse économique des conventions*, p. 13).

occupent une place importante. Dans l'introduction de l'ouvrage fondateur de l'économie des conventions, les auteurs précisent collectivement que "le terme de convention, qui désigne le dispositif constituant un accord de volonté tout comme son produit, doté d'une force normative obligatoire, doit être appréhendé à la fois comme le résultat d'actions individuelles et comme un cadre contraignant les sujets" (Jean-Pierre Dupuy et *alii*, 1989, p. 143). Dans le même ouvrage, Robert Salais détaille ce qu'il entend par convention : "Une convention, c'est un ensemble d'éléments qui, à tout instant, pour les participants à la convention, vont ensemble et sur lesquels, par conséquent, ils partagent un commun accord. Mais ce commun accord ne procède pas d'un accord préalable où chaque détail serait explicité et qui refléterait, de part et d'autre, une rationalité et une intentionnalité identique et consciente d'elle-même. Une convention est un système d'attentes réciproques sur les compétences et les comportements, conçus comme allant de soi et pour aller de soi" (Robert Salais, 1989, p. 213).

La convention relève à la fois de l'accord et de la contrainte. Comme toute règle, c'est un "dispositif cognitif collectif" (Olivier Favereau, 1989), qui génère des économies de savoir et de temps : une fois la convention connue, celle-ci permet aux agents de choisir le comportement requis en fonction des circonstances sans avoir à déterminer toutes les caractéristiques de ce comportement ou sans avoir à négocier une solution avec tous les autres agents (Eric Brousseau, 1993, p. 46-47)²¹. La convention a comme particularité de rester au niveau de l'accord tacite ; à la différence d'un contrat, elle n'a pas besoin d'être "mise en mots" pour exister (Nicolas Postel, 2000, p. 224-225). Analysant l'économie des services, Jean Gadrey montre toute l'importance des "conventions de services" entendues comme "un "climat" constitué d'attentes réciproques largement tacites, de savoirs partagés, et de règles de comportement non contractuelles c'est à dire implicites" (Jean Gadrey, 1994 C, p. 142). Bien qu'étant un arrangement socialement construit, une convention apparaît comme "naturelle"²² et non comme le produit direct d'une volonté (ou d'un accord de volontés réalisé sous la forme

²¹ De ce point de vue, une convention est une architecture de traitement spécifique de l'information collective, qui contribue à augmenter la capacité d'action d'individus dont les capacités de traitement de l'information sont limitées : une convention résume du savoir et permet à des agents n'en disposant pas d'adopter les comportements qu'ils suivraient s'ils maîtrisaient ces connaissances.

²² Même si une convention est "une régularité qui a sa source dans les interactions sociales", elle se "présente aux acteurs sous une forme objective" (Jean-Pierre Dupuy et *alii*, 1989, p. 134).

d'un compromis explicite)²³, ce qui lui confère stabilité et légitimité : les acteurs ont peu de raison d'en changer ("elle va de soi"), et une éventuelle modification leur semble hors de portée, au moins tant que la convention est "stabilisée".

Olivier Favereau distingue deux types de conventions jouant des rôles différents dans la coordination économique. Il définit d'une part, des *conventions de type 1* qui fonctionnent comme un "adjuvant au fonctionnement des marchés" grâce à leur "pouvoir d'harmonisation des anticipations" (1986, 251), dont il prend pour exemple l'analyse de Keynes des conventions qui permettent de "brider" les effets de composition existants sur les marchés financiers : "Ces conventions coordonnent non pas les décisions elles-mêmes, mais les représentations sur la base desquelles sont prises les décisions" (idem, p. 256) ; elles fournissent "un palliatif aux problèmes de la coordination par le marché, qu'elles contribuent donc à rendre plus viable et plus fiable" (idem, p. 257). Il définit, d'autre part, des *conventions de type 2* qui coordonnent cette fois les décisions elles-mêmes et qui sont un "substitut au fonctionnement des marchés, grâce à leur pouvoir d'harmonisation des comportements" (idem, p. 251), et dont l'exemple choisi sont les coutumes qui régissent les "marchés internes" du travail analysés par Peter G. Doeringer et Michaël J. Piore. Dans ce cas, la convention constitue une "alternative à la coordination par le marché, qu'elle contribue donc à cantonner dans des limites déterminées" (idem, p. 261) : "il ne s'agit plus de faire mieux fonctionner le marché mais de faire quelque chose qui fonctionne mieux que le marché" (idem, p. 257).

Une rupture avec les conventions "à la Lewis"

Les conventions ainsi analysées constituent une rupture avec l'existence des conventions "à la Lewis", perçues uniquement comme une nécessité logique pour assurer la convergence vers un équilibre (parmi plusieurs alternatives rationnelles) de comportements indépendants. Dans la vision de David K. Lewis, strictement cognitiviste et s'inscrivant dans le paradigme de la théorie des jeux, les conventions sont parfaitement arbitraires et ne procèdent de rien de plus que de "l'hyperrationalité" des individus et de leur intérêt commun à se coordonner. Elles sont le produit exclusif d'une convergence des anticipations des agents basées sur leurs croyances quant au comportement des autres. La définition de la convention de David K. Lewis (1969) est la suivante : "Une régularité R dans l'action ou les croyances est une

²³ Ceci est renforcé par le fait qu'une des caractéristiques essentielles des conventions par rapport aux autres règles est "l'impossibilité d'isoler une décision consciemment fondatrice" d'une convention (Olivier Favereau, 1986, p. 252).

convention dans une population P si et seulement si dans P les six conditions suivantes sont vérifiées : (1) Tous se conforment à R. (2) Tous s'attendent à ce que les autres se conforment à R. (3) Cette croyance que les autres se conforment à R donne à tous une bonne et décisive raison de se conformer soi-même à R. (4) Il y a une préférence générale pour la conformité générale à R. (...) (5) R n'est pas la seule régularité possible remplissant les deux dernières conditions. Il existe au moins une autre régularité R' telle qu'une croyance en le fait que les autres suivent R' donnerait à tous une bonne et décisive raison, pratique ou épistémique, de se conformer à R'. (...) (6) Les conditions (1) à (5) font l'objet d'une connaissance commune (*Common Knowledge*) qui (...) assure la stabilité". Cette dernière condition est particulièrement exigeante : en effet, l'hypothèse de "connaissance commune" recouvre beaucoup plus de présupposés que l'existence d'une simple connaissance mutuelle. Elle suppose non seulement qu'il existe un savoir partagé par les différentes personnes (degré 1), mais qu'en plus chacun sache que les autres connaissent ce savoir partagé (degré 2) et ainsi de suite avec des niveaux d'emboîtements qui doivent tendre vers l'infini²⁴. L'hypothèse de connaissance commune nécessite des acteurs dotés d'une "capacité hyperrationnelle" (Pierre Livet et Laurent Thévenot, 1994, p. 143) et on peut se demander comment peut s'établir cette connaissance commune. Cette hypothèse, souvent qualifiée "d'héroïque", confère une grande fragilité aux conventions ainsi définies : "dans la mesure où tout ne repose que sur les anticipations croisées, le moindre doute, qui paralyse l'action d'un seul suffit à faire écrouler la coordination, tel un château de cartes" (Robert Salais, Michael Storper, 1993, p. 50).

Deux caractéristiques fondamentales des conventions

A l'inverse, deux caractéristiques des conventions, telles qu'elles sont appréhendées par les auteurs conventionnalistes, expliquent ce qui fait leur force et permettent de comprendre comment elles peuvent émerger sans recourir à "l'échafaudage du *common knowledge*" (Pierre Livet et Laurent Thévenot, 1994, p. 166).

La première est que la coordination assurée par les conventions n'est que relative (Laurent Thévenot, 1997, p. 223), les conventions comportant nécessairement une part

²⁴ "Anticiper la manière dont l'autre anticipe la façon dont j'anticipe la façon dont il anticipe mon raisonnement. Et ainsi de suite de manière a priori infinie" (David K. Lewis, 1969, p. 27-28). Est qualifié de logique "spéculaire", ce processus infini d'anticipations croisées de complexité croissante (la complexité étant mesurée par le nombre d'emboîtements successifs : les anticipations des anticipations des anticipations...) sur les intentions des autres.

d'ambiguïté : "les procédures de coordination qu'elles engendrent ne sont pas susceptibles d'une application mécanique. Il existe toujours une part laissée à l'interprétation en cours d'action, et dépendant des aléas de l'environnement" (André Orléan, 1994, p. 32). Cette dimension interprétative de la convention est liée à la rationalité "interprétative" des agents économiques : l'action ne se réduit pas à une délibération a priori dont le résultat est le choix d'un plan correspondant au calcul d'un optimum, c'est un processus complexe reposant sur une tension "dialectique" entre procédures conventionnelles, antérieurement adoptées, et singularité des circonstances, tenant compte des répercussions des aléas de l'action sur les représentations. "La coordination doit être pensée comme une interprétation en cours d'action et non pas seulement comme un accord, une décision, un jugement préalables" (Pierre Livet et Laurent Thévenot, 1994, p. 145).

La deuxième caractéristique est que les conventions ont des points d'ancrages extérieurs à la seule confrontation des rationalités calculatoires individuelles, qui permettent "d'arrêter les spéculations, et de faire converger la régression des justifications" (Laurent Thévenot, 1989, p. 171). La connaissance commune de la définition lewisienne est remplacée par "une forme affaiblie de savoir collectif (...) une référence "collectivement reconnue" qui tire son efficacité du fait qu'elle arrête temporairement la logique spéculaire de mise à l'épreuve des intentions des autres" (André Orléan, 1994, p. 26). Ce savoir collectif est constitué par un ensemble de repères dont les principaux sont des "objets qualifiés". En effet, la réalisation de l'action comporte l'engagement "d'objets" qui constituent un environnement commun, pouvant permettre d'éviter de s'interroger directement sur l'action de l'autre. Ces objets peuvent servir de repères, qui renvoient notamment à des actions et à des situations antérieures, mais dont "l'objectivité" n'est pas absolue. Elle résulte "d'épreuves" correspondant à des situations de crise, d'inquiétude, de demande de garanties supplémentaires, qui vont s'arrêter par un jugement qui vise à être collectif : "dans l'épreuve, on réévalue les objets qui importent, on relève la pertinence de nouveaux indices en les constituant en objets, ce travail d'interprétation étant encadré par un mode de qualification" (Pierre Livet et Laurent Thévenot, 1994, p. 161-162). Pour remettre en cause cette qualification, il faudra prendre appui sur un autre mode de qualification. Il existe en effet plusieurs modes de qualification qui correspondent aux diverses "justifications", les individus pour se coordonner ayant à leur disposition une pluralité de registres de valeurs (Luc Boltanski, Laurent Thévenot, 1991).

De même, André Orléan montre que pour lever l'indétermination résultant du jeu des anticipations croisées, il est nécessaire qu'existe un repère collectivement reconnu qui va jouer

le rôle de point focal permettant la convergence des anticipations et l'émergence d'une communauté d'interprétation et d'action. Il souligne que la définition de ce point focal, repose moins sur la mobilisation de performances cognitives *individuelles* que sur l'existence d'une production cognitive *collective*, qui dépend certes de l'identité des individus qui participent à l'interaction, mais surtout de la manière dont ces individus se représentent l'interaction en tant que telle (André Orléan, 1994, p. 22-23). Ceci correspond à une vision d'une rationalité comme étant "située" ou "en contexte", dans laquelle interviennent les circonstances (ou éléments contextuels), mais aussi "certaines données, sociales, historiques, culturelles ou ethniques, qui permettent de faire face à l'incertitude stratégique" (Robert Boyer et André Orléan, 1994, p. 222).

Les représentations et les valeurs des acteurs ont donc un rôle important dans les conventions, l'intérêt n'étant qu'une des modalités possibles de la constitution des conventions. L'économie des conventions possède d'importantes dimensions *sociales* - avec la conception d'un "collectif qui ne se réduit pas à de l'interindividuel" (Olivier Favereau, 1994) - et *historiques*, notamment l'importance accordée à l'histoire de l'action commune. A côté de son entrée économique, elle comporte une entrée sociologique fondée sur "les exigences d'une méthode compréhensive" (Olivier Favereau), le texte fondateur des auteurs conventionnalistes précisant qu'ils entendent porter "attention à la compréhension et aux attentes des acteurs" (Jean-Pierre Dupuy et *alii*, 1989, p. 142).

C'est en tout cas, à notre avis, une dimension essentielle de l'idée de convention, même si dans un bilan, peut-être exagérément critique, Nicolas Postel estime que l'attention des principaux auteurs conventionnalistes s'est progressivement centrée sur les aspects techniques de la coordination et a conduit les conventionnalistes à des préoccupations proches des sciences cognitives, en ramenant le débat sur le terrain de la rationalité instrumentale. Selon cet auteur, cette restriction du champ d'étude aux aspects exclusivement cognitifs de la coordination a eu pour effet de mettre de côté les interrogations sur le sens de l'action et sa dimension créative (Nicolas Postel, 2000, p. 353)²⁵.

²⁵ Cette critique rejoint celle formulée par Thomas Coutrot d'une "insuffisante prise en considération de la dimension historique et conflictuelle de la coordination économique" (1998, p. 11) et celle de Christophe Ramaux, qui critique la définition de la convention comme étant "le produit d'un accord de volonté", position qui, selon lui, s'explique par la volonté de ne pas "renoncer aux préceptes de l'individualisme méthodologique" (1996, p. 189).

B - UNE TYPOLOGIE DES MONDES DE PRODUCTION BASEE SUR LES TRAVAUX DE SALAIS ET STORPER

Le cadre théorique exposé précédemment, sert de fondement à une typologie des activités économiques et des modes de coordination possibles, en termes de mondes de production. Cette typologie a été construite par Robert Salais et Michael Storper (1993), pour appréhender, au niveau macro-économique, la pluralité des modèles de systèmes productifs qui peuvent coexister au sein d'une économie nationale (1). Elle a été utilisée par d'autres auteurs pour rendre compte de la profonde diversité observée dans certains secteurs, notamment des services. L'utilisation de cette grille d'analyse, qui nous est apparue particulièrement féconde pour l'étude de l'économie du logiciel, nous a conduit à proposer quelques modifications, dont la portée est peut-être plus générale (2).

1 - La typologie de Robert Salais et Michael Storper

Leur objectif est de "prendre de front les caractéristiques actuelles d'*inégalité*, de *diversité* et d'*hétérogénéité* des situations économiques et d'enquêter sur l'existence de *plusieurs mondes possibles de production* et sur les modalités, aujourd'hui en mouvement, de leurs réagencements en multiples *mondes réels de production* (...). Ce changement de perspective, tourné vers la pluralité concrète et durable des formes de coordination économique, est, pour nous, seul à même de rendre compte des tensions persistantes de l'économie, mais aussi d'offrir un cadre de réflexion susceptible de les maîtriser" (Robert Salais, et Michael Storper, 1993, p.12).

D'autres analyses de la diversité des organisations productives existent. On peut citer, entre autres, l'*approche "relativiste"* de l'organisation de P.R. Lawrence et J.W. Lorsch (1989) pour laquelle le type d'organisation "varie suivant la nature des biens ou services fournis, les moyens mis en œuvre et l'environnement économique à un moment donné", les travaux de Henry Mintzberg (1982 et 1990) dans lesquels des *facteurs de "contingence"*, internes et externes, déterminent des mécanismes de coordination différents des activités des membres de l'organisation, dont émergent des types de configurations structurelles, ou, dans un registre différent, la mise en évidence de "*régimes de production*" différenciés, centrés sur les différentes modalités de gestion des rapports entre le travail et le capital (Michael Burawoy, 1985). L'originalité de la démarche de Robert Salais et Michael Storper est qu'elle est centrée

sur l'*identité du produit*, analysée comme le produit des conventions existantes entre l'ensemble des acteurs économiques concernés, producteurs mais aussi utilisateurs.

La pluralité effective, dans l'espace et dans le temps, des formes d'ajustement réciproque pour réduire la complexité et l'incertitude, renvoie à la pluralité des systèmes de valeurs, des justifications, des "grandeurs", des "principes de qualification", ou des "principes de justice"²⁶ auxquels les acteurs peuvent se référer de manière légitime. Elle débouche sur l'existence de "formes différentes et souvent contradictoires d'efficacité, en fonction de buts, de ressources ou de critères eux-mêmes divers" (Jean Gadrey, 1996 A, p. 269).

C'est la nature différente des formes d'incertitude auxquelles sont confrontés le producteur et le demandeur et surtout les conventions différentes qui sont requises pour que, par-delà l'incertitude les "acteurs portent à la réalité le produit dans le cours de *situations d'action* économiques au sein desquelles ils se coordonnent" (Robert Salais, et Michael Storper, 1993, p.13), qui définissent les mondes possibles de production : "il existe plusieurs mondes possibles de production, chacun centré sur un type de produit. Changer de produit, c'est se déplacer vers un autre monde possible de production. (...) Chaque monde possible de production apparaît comme un schéma de coordination entre les personnes mobilisées autour du produit : celles qui le fabriquent et celles qui l'utilisent" (idem, p.31).

Cette conception donne à l'acte de production une réalité sociale : "ce sont véritablement les acteurs économiques qui *font* le produit et non les forces exogènes (technologies et marchés) habituellement considérées par les théories économiques" (idem, p. 13). Le produit y apparaît comme une construction sociale, résultant d'un processus de coordination incluant les utilisateurs. Cette construction sociale repose sur une configuration particulière de conventions entre acteurs visant à affronter au mieux un type d'incertitude. La diversité des incertitudes implique qu'il peut exister des principes d'organisation et d'action différents, à la base des différents "mondes", chacun des mondes étant cohérent à la fois sous l'angle des stratégies ou attentes des acteurs et sous l'angle des principes d'optimisation qui le gouvernent.

²⁶ Dans un article de 1998, Robert Salais esquisse une correspondance entre les quatre mondes possibles de production et les quatre modèles de justice de Rawls (1998 B, p. 209). Il évoque le "rapprochement qu'on peut établir sans trop d'arbitraire (...) [qui] suggère que la coordination selon un monde mobiliserait, dans les attentes de chacun, une forme de référence au principe de justice correspondant" (idem, p. 277).

Par delà la conception "d'un mode de coordination économique entre les personnes, d'un type de produit et d'un registre élémentaire d'action" (Robert Salais et Michael Storper), la notion de "monde de production" peut être vue comme un modèle ou "système interprétatif" doté d'une certaine cohérence. Elle peut être considérée comme un "concept typologique" (Jean-Claude Passeron, 1991), c'est à dire une abstraction imparfaite, flexible, dont la définition formelle, toujours provisoire et nécessairement incomplète, ne suffira jamais à épuiser le sens²⁷, un type-idéal dont la signification provient du fait que des situations historiques concrètes peuvent être éclairées par ce concept, et qu'il peut servir de repère robuste dans l'investigation et l'interprétation.

Les mondes possibles de production sont conçus comme autant de registres d'action, de conventions constitutives, qui permettent aux personnes de faire face à l'incertitude radicale qui entoure leurs activités. La typologie des mondes de production est résumée dans le tableau XXXIX (Robert Salais, Michael Storper, 1993, p.43). Elle est construite à partir de deux critères : d'une part une distinction entre des "produits dédiés" et des "produits génériques", d'autre part une distinction entre des "produits spécialisés" et des "produits standardisés", la notion de spécialisation faisant référence à un travail de professionnels d'experts, par opposition à un produit standard où le producteur n'a "aucune possibilité de signifier son identité sur le produit" (Robert Salais, 1994, p.385). Le croisement de ces deux critères aboutit à l'existence possible de quatre mondes, qui renvoient à "quatre modèles de qualité du produit", cette qualité étant elle-même appréhendée en combinant les deux domaines de la production et de l'échange.

²⁷ A la différence des "concepts logico-empiriques" que l'on peut définir de façon "pure" avec la rigueur formelle des sciences expérimentales ou logico-mathématiques.

Tableau XXXXIX : Les mondes de productions selon Robert Salais et Michael Storper

		PRODUITS SPECIALISES	PRODUITS STANDARD		
PRODUITS DEDIES	Evaluation de la qualité :	<i>Le prix</i>	Evaluation de la qualité :	<i>Standard particulier à la situation</i>	INCERTITUDE
	Forme d'incertitude :	<i>Incertitude vis-à-vis de l'autre</i>	Forme d'incertitude :	<i>Méconnaissance du futur immédiat</i>	
	Traitement :	<i>Compréhension au sein d'une communauté de personne</i>	Traitement :	<i>Disponibilité immédiate</i>	LE DEMANDEUR
	Concurrence :	<i>Qualité</i>	Concurrence :	<i>Prix, qualité</i>	
		LE MONDE DE PRODUCTION INTERPERSONNEL	LE MONDE DE PRODUCTION MARCHAND		
PRODUITS GENERIQUES	Evaluation de la qualité :	<i>Règles éthiques et scientifiques</i>	Evaluation de la qualité :	<i>Standard général</i>	PREVISIBILITE
	Forme d'incertitude :	<i>Incertitude sur le futur</i>	Forme d'incertitude :	<i>Risque probabilisable</i>	
	Traitement :	<i>Confiance envers l'autre</i>	Traitement :	<i>Prévision à court et moyen terme des événements et des comportements</i>	LE DEMANDEUR
	Concurrence :	<i>Par l'apprentissage</i>	Concurrence :	<i>Prix</i>	
		LE MONDE DE PRODUCTION IMMATERIEL	LE MONDE DE PRODUCTION INDUSTRIEL		
		Economie de variété	Economie d'échelle		
LE PRODUCTEUR					

2 - Quelques propositions d'amendements

L'utilisation répétée dans l'économie des services de cette grille d'analyse, qui a confirmé ses capacités heuristiques pour étudier des secteurs aussi divers que le tourisme, la Poste, les services de soutien aux personnes âgées, les prestations de l'ANPE..., nous²⁸ a conduit à formuler des précisions et propositions de modifications, qui dépassent peut-être les adaptations légitimes aux spécificités des secteurs étudiés (cf. notamment Jean Gadrey, 1996 B).

Un premier problème concerne l'application aux services d'un cadre construit principalement par rapport aux secteurs industriels. Certes, Robert Salais et Michael Storper indiquent qu'ils utilisent le terme de "produit" et non celui de "bien", parce qu'ils jugent "assez largement caduque l'opposition traditionnelle entre les biens (...) et les services, (...) tout produit incorporant un service" (1993, p. 58-59). Ils précisent même qu'un "produit peut être un pur service d'une personne à une autre" (idem, p. 59) mais les références utilisées pour illustrer la pertinence des quatre modèles appartiennent au secteur industriel : les entreprises de production industrielle de masse pour le monde de production industriel, les entreprises organisées selon les principes de flexibilité, capables de produire du "sur mesure industriel" pour le monde de production marchand, les districts industriels locaux pour le monde de production interpersonnel, et les firmes industrielles de haute technologie pour le monde de production immatériel (Jean Gadrey, 1996 A, p. 39-40). Or, l'intégration dans cette grille des activités de services n'est pas sans poser quelques difficultés : par exemple, des services peu "spécialisés" bien que non standard, comme les services d'aide à domicile, sont difficiles à classer dans les mondes ainsi définis.

A notre avis, ces difficultés proviennent notamment d'ambiguïtés dans l'application du critère "produits spécialisés / produits standard". Il semble qu'en fait la véritable distinction utilisée par Robert Salais et Michael Storper concerne davantage le degré de spécialisation ou de standardisation du *travail* que du *produit*. Ce glissement semble justifié par le raisonnement suivant de Robert Salais : "les diverses résolutions de l'incertitude des interactions ont un point fixe : l'acceptation du produit final par un demandeur. C'est l'épreuve de réalité qui dénoue l'incertitude et arrête pour chacun la quête d'informations sur ce qu'il fait

²⁸ Le "nous" renvoie ici au groupe "Services, innovation, évaluation" du CLERSE.

et sur ce que font les autres. (...) Aussi, si nous sommes capables de dégager les modalités possibles de cette épreuve, ne pouvons-nous, par une sorte de remontée du raisonnement (*backward induction*), délimiter les formes correspondantes de traitement de l'incertitude dans le cours, "antérieur", de l'action de travail et, ainsi, les conventions de travail mises en œuvre ?" (1994, p. 373). Or ce raisonnement peut certes s'appliquer au travail direct incorporé dans un produit (considérer qu'un produit standard a été effectué avec du travail standardisé), mais pas nécessairement au travail indirect (un produit non standard peut receler indirectement du travail standard qui a servi à produire des composants de ce produit). Cette question renvoie peut-être à une perception de la standardisation trop limitative : pour Robert Salais et Michael Storper, la standardisation semble conduire nécessairement à la production de produits standard et exclure les possibilités de différenciation. Certes, historiquement, le processus de standardisation avait pour conséquence inéluctable la standardisation des produits dont l'intérêt (économie d'échelle, effets d'apprentissage améliorant la productivité) était qu'elle reposait sur une standardisation des composants et des méthodes de production mais dont l'inconvénient était l'uniformité des produits (satisfaction de besoins standard ou nécessité d'une adaptation de l'utilisateur au produit standard) ; toutefois, il existe maintenant d'autres processus de standardisation permis notamment par les innovations techniques et organisationnelles qui combinent standardisation des composants²⁹ et des méthodes de production (facteurs d'efficacité) et différenciation des produits (adaptation du produit au besoin précis de l'utilisateur).

A notre sens, c'est cette vision limitative de la standardisation qui explique les difficultés pour identifier le monde marchand - il n'est pas évident d'imaginer ce qu'est un produit à la fois "standard" et "dédié" - et pour le distinguer du monde industriel. La différence avec le monde de production industriel paraît résider dans la question des délais de production ("disponibilité immédiate") sans pouvoir appréhender les caractéristiques différentes des produits fabriqués dans chacun de ces mondes, ce qui se traduit dans les hésitations sur l'expression choisie pour résumer l'évaluation de la qualité dans le monde marchand : le "standard particulier à la situation" (Robert Salais et Michael Storper, 1993, p.

²⁹ A côté de la standardisation des composants, facteur d'économie d'échelle, il faut également prendre en compte la question de la standardisation des interfaces (entre différents produits, entre le produit et l'utilisateur) qui jouent un rôle de plus en plus important pour faciliter l'utilisation et l'intégration de produits différenciés.

43) est devenu "standard industriel choisi par le demandeur" dans l'article de Robert Salais (1994, p. 386).

La dernière remarque concerne le choix des termes pour désigner les différents mondes de production. Certes, le caractère relativement arbitraire de cette opération est inévitable, aucun adjectif ne pouvant résumer seul les différentes dimensions et la richesse d'un monde de production. Toutefois, il nous semble que l'adjectif "marchand" ne saurait qualifier le seul monde de production "standard-dédiés", alors que la dimension marchande est largement présente dans d'autres mondes de production. De même, la dénomination de "monde de production immatériel" apparaît peu appropriée pour désigner la singularité des activités de création, alors que des productions immatérielles sont par ailleurs réalisées dans d'autres cadres : par exemple, le monde de production "industriel" englobe également une partie non négligeable des activités de services, alors que les services sont souvent identifiés par une certaine "immatérialité".

Par rapport à ces difficultés, nous proposons de construire une typologie amendée sur les bases suivantes. Les mondes de production se différencient selon le type de produits. Leur fondement reste inchangé, à savoir les formes différenciées de traitement de l'incertitude, celles-ci étant simplement séparées sur le plan analytique, selon qu'elles visent à répondre à l'incertitude du client-usager sur la qualité du produit fourni (ce qui renvoie à une incertitude sur la qualité du processus de production mais aussi à une incertitude sur la compréhension des attentes du client-utilisateur par le producteur), ou à l'incertitude du producteur sur les comportements actuels et futurs (demande anticipée) des clients – utilisateurs. Le premier critère utilisé reste celui de Robert Salais et Michael Storper, produits dédiés / produits génériques, en insistant sur ses conséquences concernant la relation producteurs / clients - usagers ; à une quasi-absence de relation caractéristique d'une production "*poussée*" vers le marché ou les usagers, sont opposées les situations où la production est "*tirée*" par la demande. Le deuxième critère devient l'absence ou l'existence de phénomènes de standardisation dans la fabrication du produit, en soulignant que la standardisation ne s'étend pas nécessairement au produit lui-même mais peut se limiter aux méthodes et aux composants utilisés.

Il en découle des différences selon les mondes de production, mondes que nous avons "rebaptisés", et spécifiés précisément au regard de l'évaluation *ex-ante* de la qualité du produit, des modalités de la concurrence entre producteurs, des modes de coordination entre producteurs et utilisateurs, des modes de rationalisation de l'activité, et du type dominant d'entreprise. La typologie proposée est résumée dans le tableau L :

Tableau L : La typologie modifiée des mondes de production.

	Absence de standardisation	Standardisation du produit et/ou des composants
Produits dédiés	Monde interpersonnel	Monde de la production flexible
	Type de produit : <i>Sur-mesure ; service personnalisé</i>	Type de produit : <i>Des gammes de produits diversifiés livrées en juste à temps jusqu'au sur-mesure de masse</i>
	Forme d'incertitude : <i>Incertitude réciproque sur les besoins précis et sur la qualité du produit</i>	Forme d'incertitude : <i>Incertitude réciproque sur les besoins précis et sur la qualité du produit</i>
	Traitement par rapport au produit : <i>Intercompréhension, métier, qualités professionnelles et relationnelles</i>	Traitement par rapport au produit : <i>Formes plus ou moins développées de coproduction</i>
	Traitement par rapport à la demande future : <i>Adaptation ; barrières institutionnelles</i>	Traitement par rapport à la demande future : <i>Flexibilité dynamique</i>
	Evaluation de la qualité : <i>Confiance, réputation, qualification</i>	Evaluation de la qualité : <i>Certification de l'organisation ou des processus de production</i>
	Concurrence : <i>Qualité</i>	Concurrence : <i>Qualité et prix</i>
	Mode de coordination : <i>"Marché-jugement", hiérarchie, réseau "social"</i>	Mode de coordination : <i>Marché d'organisations, marché segmenté, réseau "économique"</i>
Mode de rationalisation : <i>Economies de variété, expérience professionnelle</i>	Mode de rationalisation : <i>Economies d'échelle et économies de variété</i>	
Type dominant d'entreprise : <i>Travailleurs indépendants, P.M.E. ou départements internes d'entreprise</i>	Type dominant d'entreprise : <i>Entreprises en réseau, firmes J</i>	
Produits génériques	Monde de la création	Monde fordiste
	Type de produit : <i>Création originale ; biens intangibles</i>	Type de produit : <i>De masse, standardisé</i>
	Forme d'incertitude : <i>Radicale (sur l'émergence du produit et son utilité)</i>	Forme d'incertitude : <i>Faible, risque probabilisable</i>
	Traitement par rapport au produit : <i>Règles éthiques et scientifiques ; critères esthétiques</i>	Traitement par rapport au produit : <i>Besoin standard</i>
	Traitement par rapport à la demande future : <i>Financement principalement non marchand</i>	Traitement par rapport à la demande future : <i>Prévision, action sur la demande et flexibilité quantitative</i>
	Evaluation de la qualité : <i>Jugement des pairs</i>	Evaluation de la qualité : <i>Standard industriel de produit</i>
	Concurrence : <i>Prestige</i>	Concurrence : <i>Prix</i>
	Mode de coordination : <i>Rôle secondaire du marché ; dans certains cas, libre disponibilité</i>	Mode de coordination : <i>Marché anonyme, de masse (grandes séries)</i>
Mode de rationalisation : <i>Recherche et inventions ; inspiration</i>	Mode de rationalisation : <i>Economies d'échelle</i>	
Type dominant d'entreprise : <i>Universités, centres de recherche (privé ou public), créateurs indépendants</i>	Type dominant d'entreprise : <i>Bureaucraties industrielles ou tertiaires "mécanistes" ; firmes A</i>	

a - Le monde interpersonnel

Le monde interpersonnel est celui de la production sur-mesure et du service personnalisé, d'une production unique et présentant toujours des différences plus ou moins importantes avec les autres produits et visant une adaptation fine aux besoins du client. Ce monde est qualifié d'interpersonnel, car les relations directes (sans la médiation d'un objet) entre les personnes, productrices et utilisatrices, jouent un rôle déterminant. A l'opposé des transactions anonymes, les échanges s'effectuent selon un mode "idiosyncrasique" où il est nécessaire que s'établisse une communication entre les deux contractants, afin qu'existe une compréhension réciproque des attentes et caractéristiques des utilisateurs, et des capacités du producteur en vue de fournir une prestation spécifique.

Ce monde englobe des activités diverses comme l'artisanat, la production de certains équipements spécialisés, des services aux entreprises très spécifiques (la majorité des activités de conseil par exemple), les services sur mesure aux personnes depuis les services peu qualifiés d'aide à domicile jusqu'aux prestations de professionnels très qualifiés (consultation médicale, conseil juridique, architecture). C'est notamment le monde des "professions consultantes" (Eliot Freidson, 1984).

L'incertitude du producteur sur la demande future peut être atténuée dans certains cas par ses capacités d'adaptation à la diversité des produits à fournir (mobilisation d'économies de variété), et parfois par l'existence de barrières institutionnelles à l'exercice de l'activité. Si la concurrence porte surtout sur la qualité du produit fourni³⁰, cette qualité est souvent difficile à évaluer surtout *ex-ante*. En effet, dans ce monde les méthodes peu formalisées, les savoir-faire empiriques, les connaissances tacites conservent une certaine importance. Ne faisant pas l'objet d'une explicitation suffisante pour être standardisés, ils ne peuvent donc pas être séparés de l'acteur et constituent donc des éléments cognitifs difficiles à communiquer. Les qualités du produit étant déterminées dans le cours de l'échange, et non préalablement à l'échange, un contrat ne peut être qu'incomplet : non seulement il est impossible de tout spécifier, mais dans certains cas la recherche d'une trop grande précision risque d'engendrer le soupçon et la méfiance des partenaires, alors même que le but recherché est de susciter la confiance (Pierre Livet et Laurent Thévenot, 1994, p. 155-156). De ce fait, le client-utilisateur

³⁰ Comme il n'existe pas une définition standard complète du produit échangé, le prix ne peut être l'unique lien entre les contractants.

doit fréquemment se rabattre sur des "signes" indirects qui servent d'indices pour fonder un jugement sur le produit. Ces signes peuvent être des signaux visibles comme la qualification dûment reconnue du prestataire, constituant une garantie relative de ses compétences professionnelles, ou des signaux non visibles comme la réputation.

On se trouve dans une situation qualifiée par Lucien Karpik (1989) de "marché-jugement" où la confiance peut s'instaurer "grâce à des réseaux sociaux fondant la bonne réputation de l'entreprise" (Michel Callon, 1999, p. 22) : le jugement repose sur la collecte de renseignements auprès de tierces personnes appartenant à un "réseau échange" et auxquelles le client accorde sa confiance. Le mécanisme réputationnel dérive de l'existence d'un "être social" fondé sur "l'ensemble des liens sociaux³¹ qu'il a accumulés et qui constituent son identité sociale" (André Orléan, 1996, p. 19)".

La nécessité de fonder les relations sur la confiance et l'intercompréhension explique qu'elles acquièrent fréquemment un caractère durable, fondé sur une bonne connaissance mutuelle et des comportements de fidélité. Ceci permet également de comprendre pourquoi dans certains cas, notamment dans des services aux entreprises, cette activité est assurée de façon interne, l'importance des coûts de transaction et la spécificité des actifs mobilisés (au sens de Williamson) n'étant pas compensée par la faiblesse des économies d'échelle possibles.

b - le monde fordiste

A l'opposé de ce premier monde se trouve le monde fordiste. Ce monde, abondamment analysé, est celui de la production de masse (production de biens ou de services standardisés en très grande série), destinée à des marchés étendus et composés de demandeurs (clients ou usagers voire "administrés" quand la production est publique) considérés comme anonymes.

Si c'est d'abord dans les secteurs industriels où il s'est implanté initialement que le fordisme a été le plus étudié, des caractéristiques déterminantes de ce monde se retrouvent également dans des activités de services aussi différentes que la restauration rapide, le transport, la partie la plus importante de l'activité bancaire et assurancière, les services postaux mais également le tourisme de masse. Les échanges s'effectuent alors à travers des

³¹ André Orléan démontre que ce qui permet à ces liens sociaux (voisinage, parenté, métier,...) d'être des gages crédibles dans l'établissement de la confiance, c'est qu'ils se situent dans "un espace radicalement distinct du calcul et de l'intérêt". L'"incomplétude de la logique marchande pure" ne peut être surmontée que par l'existence "de formes d'appartenance échappant à l'ordre économique" (1996, p. 7 et 20).

marchés dont la particularité réside dans le fait que les prestations sont relativement indépendantes des personnes qui les fournissent, avec une relation de service limitée, tendance que renforce le développement de l'usage de systèmes automatiques de libre service.

C'est dans ce monde que l'incertitude est la plus faible grâce à la standardisation du produit (industriel ou de service) ; de ce fait, l'utilisateur connaît assez précisément les caractéristiques du produit et celui-ci répondant à des besoins standards (les demandeurs sont considérés comme identiques), le producteur peut anticiper l'évolution de la demande, l'adaptation à une évolution conjoncturelle plus instable étant réalisée par une flexibilité quantitative. L'utilisateur a une influence minimale sur les caractéristiques du bien ou du service : en appliquant les catégories de Albert O. Hirschman (1972), on peut dire que ses possibilités de *voice* sont réduites, et que la portée de l'*exit* est limitée dans la mesure où la concurrence, quand elle existe, porte principalement sur le prix avec une faible différenciation des produits, l'objectif principal étant de produire à moindre coût en respectant des standards.

Les producteurs sont de grandes entreprises générant des gains de productivité à partir des économies d'échelle obtenues par la production en très grandes séries. C'est le monde de la bureaucratie mécaniste décrit par Henry Mintzberg (1982) en des termes proches de ceux avec lesquels Masahiko Aoki (1986) définit la "firme A" (américaine) ou "modèle H" (1991). Ce modèle de production s'est révélée efficace dans un environnement relativement simple et plutôt stable, et dont les principales caractéristiques, résumées par Jean Gadrey (1994 A), sont : le travail opérationnel de base y est réglé selon des procédés de travail fortement standardisés et spécialisés ; la conception de l'organisation, de la standardisation, des tâches et de leur contrôle (très développé, parfois de façon obsessionnelle) est le fait de "fonctionnels" spécialisés (la technostructure) ; la ligne hiérarchique est développée ; l'organisation utilise en général de façon privilégiée la quantification (des objectifs, des résultats, des ventes, des pièces, des temps...) comme mode de mobilisation-contrainte-contrôle des individus.

c - le monde de la création

A côté du monde fordiste existe un autre monde dominé par l'offre où le client-utilisateur a peu d'influence sur la nature de la production effectuée : le monde de la création. C'est dans ce monde que s'effectuent des créations originales d'œuvres nouvelles de nature très diverses (scientifiques, artistiques...) : la partie des services qui obéissent à une "logique de représentation humaine ou spectacle vivant" (Jean Gadrey, 1999 A), des biens tangibles non reproductibles comme les œuvres d'art, des "originaux" constitutifs de biens intangibles

(texte, film...)³², le monde de la recherche, du développement des connaissances fondamentales par opposition à la recherche appliquée...

Si d'un point de vue économique, ce monde pouvait apparaître comme un peu marginal par rapport au reste de l'activité économique, c'est (et ce sera) de moins en moins le cas en raison de deux facteurs. Tout d'abord, la part de plus en plus importante prise par cette production, notamment des biens intangibles, dans la consommation des ménages. D'autre part, l'importance cruciale de la recherche fondamentale et des innovations techniques radicales, sans lesquelles "le progrès technique finirait par disparaître" (Chris Freeman, 1988, p. 36). Or les retombées – qui ne sont pas toutes volontaires - des programmes de recherche sont difficiles à estimer *ex-ante*, surtout quand il s'agit de technologies génériques, où l'incertitude concerne la distribution de la variété produite par la technologie, les caractéristiques de la population des variantes, des types, des dérivés qui seront engendrés par sa dynamique (Dominique Foray, 1996, p. 261)³³.

Ceci n'est qu'une des facettes de l'incertitude radicale, constitutive de ce monde. C'est en effet dans le monde de la création que l'incertitude est la plus importante. Elle englobe l'émergence même d'un produit au terme du processus de création, son aptitude à satisfaire des besoins actuels ou futurs à un coût acceptable, et la capacité du producteur à bénéficier du fruit de sa production. De façon à peine caricaturale, le monde de la création est constitué d'œuvres à la recherche d'un public, de solutions à la recherche de problèmes, d'inventions à la recherche d'utilisations.

D'une certaine façon, on peut dire qu'une des particularités du monde de la création est d'assumer l'incertitude et non de chercher à la réduire. La conséquence directe est que, dans la plupart des cas, l'activité ne peut s'effectuer que si le marché ne joue pas un rôle déterminant : la production peut être effectuée par une unité économique non marchande (université, centre de recherche public), ou par un département de l'entreprise non soumis à une contrainte de rentabilité à court terme ; quand la production est marchande, son succès économique n'est pas jugé comme étant le critère essentiel de la réussite (création artistique, littéraire) et n'est

³² Les biens intangibles, qui sont à l'inverse très facilement reproductibles (à un coût qui avec l'évolution technologique tend vers zéro), se caractérisent par le fait que la copie et l'original sont strictement identiques (cf. chapitre III).

³³ De même, Giovanni Dosi et Luigi Orsenigo insistent sur "l'incertitude intrinsèque quant aux résultats technologiques et économiques des activités de recherche" (1988, p. 16).

pas l'unique critère d'allocation de ressources (mécénat, subvention) ; dans la plupart des cas le financement public direct ou indirect (aides) joue un rôle important.

Ceci peut s'expliquer par des caractéristiques de biens collectifs qu'ont nombre de ces créations, par rapport auxquelles des dispositifs techniques et juridiques s'avèrent inefficaces ou négatifs pour la collectivité. Mais cet effacement relatif du rôle du marché peut également être justifié par le nécessaire maintien d'une grande diversité, souvent injustement associée au fonctionnement du marché : ainsi, dans le cas de la recherche, Michel Callon montre que "la science doit être considérée comme un bien public et qu'elle doit par conséquent échapper au marché" (1993, p. 4) non pas parce qu'elle posséderait les propriétés intrinsèques des biens publics au sens de la théorie économique, mais parce que "le marché se transforme en une puissante machine à fabriquer de l'irréversibilité et à restreindre la variété des options technologiques, c'est à dire l'espace des choix possibles" (idem, p. 15)³⁴. De même, Dominique Foray note que la recherche militaire, dont la justification n'a jamais été marchande, a permis d'épargner de la "standardisation générale d'un domaine, (...) certains sous-systèmes qui constituent autant de réserves de diversité" (1996, p. 271)³⁵.

Dans le monde de la création, l'évaluation de la qualité obéit à des règles éthiques et scientifiques ou à des critères esthétiques avec une place importante accordée au jugement des pairs ; l'allocation des ressources s'effectue principalement par une évaluation de type institutionnel ; les compétences mobilisées sont les capacités à la recherche et à l'invention, le moteur de la concurrence étant le prestige résultant de l'originalité de la création. Le monde de la création en terme de motivation des acteurs renvoie au "monde de l'inspiration" analysé par Luc Boltanski et Laurent Thévenot (1991, p. 200-206). C'est un monde peu stabilisé qui correspond à la "source chaude" du foisonnement de la découverte, synonyme d'imagination et de créativité sans entrave, confronté à la "source froide" du marché et de la rigueur (Michel Callon et Bruno Latour, 1991, p. 12). Si son importance est grandissante, il est confronté en

³⁴ Nous reprendrons largement ce raisonnement, basé sur la force des rendements croissants d'adoption, pour le cas des logiciels, où il nous semble qu'il est particulièrement vérifié.

³⁵ Dominique Foray cite l'exemple des technologies de propulsion hypersonique, abandonnées par la recherche civile dès les années soixante, toujours développées par la recherche militaire et qui retrouvent actuellement une certaine actualité, y compris pour des utilisations civiles. S'il se refuse à "plaider pour le maintien des budgets de R-D militaire", il se demande "quelles institutions pourront se substituer aux programmes militaires dans ce rôle de maintien d'une certaine diversité" (1996, p. 271).

permanence aux forces puissantes qui à l'inverse poussent à l'extension sans limites du règne de la marchandise³⁶.

d - le monde de production flexible

Dernier monde (au sens de son apparition historique), le monde de production flexible s'étend de la production d'une gamme diversifiée de biens et de services standards livrés en juste-à-temps jusqu'à un véritable sur-mesure de masse (*mass-customization*) qui prend deux formes : des produits individualisés fabriqués à partir de composants et/ou de méthodes standardisés ; des composites (*compacks*), combinaisons en fonction d'un besoin individuel de biens et de services dont certains sont standardisés.

Dans sa variante la moins en rupture avec le fordisme, il s'agit simplement à partir de produits ayant des caractéristiques générales identiques, de les adapter à une demande précise de consommation en jouant sur des caractéristiques secondaires (choix au sein d'une gamme limitée d'alternatives prédéfinies) et sur l'adjonction d'options diverses devenues plus nombreuses et plus faciles à intégrer. Dans l'industrie, les secteurs automobiles ou de l'habillement sont des exemples de cette évolution que facilitent les possibilités de "différenciation retardée" des produits. Les transformations des prestations des banques et des assurances dans les années quatre-vingt-dix constituent également une illustration de cette mutation dans les services.

Une variante plus ambitieuse, pour laquelle la production apparaît comme véritablement "tirée" par la demande, consiste à réaliser un produit sans délai pour répondre à un besoin individuel, en assemblant ou en combinant des composants, des modules, des éléments qui, eux, sont standardisés. Faïz Gallouj parle d'un monde de production "flexible ou modulaire" pour insister sur l'importance de la "modularisation" qui permet de concilier logique de standardisation et logique de relation de service (1997, p. 69). La production de micro-ordinateurs par Dell, mais aussi la "cuisine d'assemblage" dans la restauration, de plus en plus de prestations touristiques, en sont des exemples significatifs.

³⁶ En d'autres termes, on peut considérer que le monde de la création, à l'inverse des autres mondes, se caractérise par la prééminence de la valeur d'usage (au sens large de ce terme, non utilitariste, intégrant par exemple des dimensions éthiques et esthétiques) sur la valeur d'échange.

Une troisième variante consiste à augmenter la "flexibilité du produit industriel" en proposant, avec un bien qui reste standardisé, la fourniture conjointe de services de plus en plus divers, qui dépassent les classiques services d'après-vente et de garantie, pour apporter une solution globale et personnalisée à un problème individuel (Jean Gadrey, 1996 A, p. 193).

Ces différentes évolutions, qui bien évidemment peuvent se combiner, correspondent à des tendances "lourdes" à l'œuvre dans la société, de "démassification", de personnalisation de l'offre et de la demande, et d'exigence croissante de procédures interactives (idem, p. 219).

On retrouve dans ce monde les mêmes formes d'incertitude que dans le monde interpersonnel avec le même impératif d'intercompréhension et donc de développement de la relation entre producteurs et clients-usagers. Si dans certains cas la production s'effectue en réponse à un besoin particulier formulé par l'utilisateur, dans d'autres cas, la demande précise de l'utilisateur se construit au cours du processus de production ; dans ces cas, *la relation de service s'étend jusqu'à la coproduction du produit en mobilisant les compétences du producteur mais également celles de l'utilisateur*. L'acheteur (individu ou entreprise) apprend à exprimer une "demande multidimensionnelle" qui comporte des éléments de plus en plus forts de "valeur d'utilisation" et d'individualisation, mais qui se situe cependant dans un référentiel informé des possibilités de la production / consommation de masse (types de produits standards, prix...) (Albert Bressand et Kalypso Nicolaidis, 1988, p. 147).

La concurrence porte simultanément sur la qualité et le prix : "l'avantage comparatif réside dans un dosage subtil entre des économies d'échelle réductrices de coût, et l'individualisation créatrice de valeur" (idem, 1988, p. 147). Par rapport au monde fordiste où la compétition reposait principalement sur le couple prix-volume, interviennent d'autres critères comme la variété de l'offre, le temps et les délais, la nouveauté, l'innovation qui supposent de fortes capacités de flexibilité et de réactivité des entreprises (Anne Mayère, 1997, p. 200).

On peut définir les ambitions des entreprises du monde de production flexible comme une tentative pour concilier les objectifs, apparemment contradictoires, du monde fordiste et du monde interpersonnel. Cette ambition implique une stratégie qui combine "la recherche d'effets de productivité et d'apprentissage classiques (volumétriques), la recherche d'effets relationnels (individualisation du produit et flexibilisation des processus) et la recherche d'effets de qualité d'une manière qui échappe à la logique artisanale du "produit bien fait"

pour relever de la technique de production et d'organisation" (Albert Bressand et Kalypso Nicolaidis, 1988, p. 147).

La combinaison d'économie de variété et d'économie d'échelle est permise à la fois par des innovations technologiques (notamment dans les technologies de l'information) et par des innovations organisationnelles. Les potentialités des nouvelles technologies de l'information ayant été analysées précédemment, notamment pour réaliser une automatisation flexible³⁷ (cf. chapitre IV), on se limitera à souligner les possibilités qu'elles apportent pour segmenter de plus en plus finement les marchés par la "télématisation des relations qui permet une définition beaucoup plus fine du client, définition qui devient un intrant du processus de production" (idem, 1988, p. 147).

Sur le plan organisationnel, de multiples expressions existent pour synthétiser les principales caractéristiques du monde de production flexible : la *firme J* de Masahiko Aoki (1991) qui se distingue par sa structure informationnelle horizontale ; un nouveau paradigme organisationnel qualifié de *néofordiste* que Fabio Arcangeli et Christian Genthon résument par le triptyque *concurrent engineering, lean production, et total quality*, en insistant sur une organisation plus horizontale avec la réduction des niveaux hiérarchiques, une plus grande adaptabilité interne et externe, et le développement de la coopération grâce à des firmes "réseau" (1997, p. 155 et 159) ; un nouveau modèle industriel caractérisé par une *flexibilité dynamique* qui permet "d'assurer un fort dynamisme technologique" et surtout "d'offrir, à chaque étape des potentialités de développement diversifiées permettant de s'adapter aux évolutions des demandes, des marchés et des technologies" (Olivier Weinstein, 1992, p. 35) ; *l'Organisation Basée sur l'Information* (OBI) dans laquelle l'information est considérée comme une ressource organisationnelle qui doit être partagée et circuler facilement pour accroître la réactivité de l'entreprise à toute modification (Paricia D. Fletcher, Lester Diamond, 1997).

Le travail dans le monde de production flexible se caractérise par la place croissante accordée à la maîtrise de processus et d'événements et au développement de la composante décisionnelle, au détriment de la conception ou de la réalisation d'objets ou de prestations

³⁷ Ehud Zuscovitch (1983) montre comment l'informatisation peut permettre de concilier "variété et efficacité" : la production peut se dérouler selon un "processus continu même avec variété considérable des produits" (p. 59) qui correspond à des "formes particulières de standardisation-différenciation" (p. 60).

isolés (Pierre Veltz et Philippe Zarifian, 1994). Le travail "renoue avec ce qui faisait l'efficacité de l'artisanat, cette intelligence globale du produit, qui permet, face aux inéluctables et permanents incidents imprévus, des ajustements d'une qualité bien supérieure à ceux que peut proposer un salarié enfermé dans un segment du processus" (Thomas Coutrot, 1998, p. 25-26). Cette "production réflexive" (M. Freyssenet, 1995) implique le passage d'une convention fondée sur le couple "subordination contre sécurité" (caractéristique du fordisme) à une convention "adhésion contre autonomie relative" (Anne Mayère, 1997, p. 209). Anne Mayère souligne que, si cette évolution permet une plus grande valorisation des salariés, en raison des responsabilités qui leur sont dédiées, des capacités d'initiative qui leur sont ouvertes et des relations qu'ils ont à développer, elle peut également être "une source de stress et fonder une subordination encore plus radicale que celle à l'œuvre dans le modèle taylorien, puisqu'elle tient au savoir-être et aux productions intellectuelles" (idem, p. 210).

En conséquence, l'appréciation et la mesure des performances ne peuvent être que globales, en dépendant notablement de la qualité des interactions et de la pertinence des règles de coordination (Pierre Veltz et Philippe Zarifian, 1994). En reprenant les expressions de Henry Mintzberg, on peut dire que la coordination par ajustement mutuel prend le pas sur la coordination par standardisation des procédures. Les interactions à prendre en compte ne concernent pas que les relations internes à l'entreprise. Interviennent également de plus en plus, la richesse des relations que l'entreprise noue avec ses différents partenaires (clients, fournisseurs, sous-traitants...) d'où la constitution de "réseaux économiques" fondés sur une coopération durable et mutuellement profitable : s'établissent des relations de partenariat instituant des groupes, des alliances ou des organisations étendues, dont la solidité est liée au coût financier et organisationnel de la rupture pour les acteurs et à la possibilité de "négociations permanentes pour maintenir la relation dans un entre-deux de confiance et d'intérêt" (Philippe Steiner, 1999, p. 90-91). Ces réseaux doivent être particulièrement robustes quand la réalisation d'assemblages complexes de biens et de services (*compacts*) dépend de l'existence de structures multilatérales permettant une véritable "multiproduction" (Albert Bressand, 1994, p. 274).

L'évaluation de la qualité de la production repose principalement sur des procédures de certification, que celles-ci soient effectuées par des institutions ou par l'entreprise cliente elle-même (cotation des sous-traitants par exemple). La certification portant le plus souvent sur la qualité des processus, elle correspond à l'existence d'un "marché d'organisation" : cette expression, par opposition à un "marché de produits", vise à souligner que l'évaluation ne

porte pas sur un produit donné, mais sur l'organisation (François Eymard-Duvernay, 1994, p. 329).

Section II - Les mondes de production du logiciel

Ce cadre d'analyse général des mondes de production est maintenant appliqué au cas particulier de la production des logiciels. Nous ne détaillerons pas pour la production des logiciels l'ensemble des caractéristiques présentées précédemment sur les mondes de production en général, en insistant simplement sur les différences pouvant exister avec le modèle général en raison des spécificités technico-économiques des logiciels. La typologie, présentée dans le tableau suivant, résume uniquement, pour chaque monde de production, la façon dont il se concrétise en un type de produit logiciel particulier, avec un exemple caractéristique, et les producteurs et utilisateurs principaux de chacun des mondes. Il importe de préciser que ces mondes de production étant basés sur des types de produits différents, ils ne sont pas nécessairement alternatifs : dans certains cas, il peut exister entre eux des relations de complémentarité, le produit d'un monde pouvant constituer un des inputs d'un autre monde³⁸. Cette typologie permet d'analyser comment les différents mondes de production du logiciel se positionnent différemment face aux trois dimensions critiques des logiciels (cf. chapitre V), à savoir la productivité dans la production des logiciels, la fiabilité des logiciels produits, et l'adéquation aux besoins des utilisateurs.

³⁸ Par contre si la production d'une même entreprise peut correspondre à plusieurs mondes différents, la logique d'un des mondes domine nécessairement en imprimant sa "marque" à l'ensemble de l'entreprise.

Tableau LI : Les mondes de production du logiciel

	Absence de standardisation	Standardisation du produit et/ou des composants
Produits dédiés	Monde interpersonnel	Monde de la production flexible
	Type de produit : <i>Logiciels sur-mesure développés "ex-nihilo"</i> Exemple caractéristique : <i>Application spécifique</i> Producteurs principaux : <i>Services informatiques internes, S.S.I.I.</i> Utilisateurs principaux : <i>Grandes entreprises, Administrations</i> Productivité dans la production des logiciels : <i>Faible. Contradictoire avec fiabilité</i> Fiabilité des logiciels produits : <i>Contradictoire avec productivité.</i> Adéquation aux besoins des utilisateurs : <i>Variable selon le niveau d'intercompréhension producteurs/utilisateurs</i>	Type de produit : <i>Logiciels standards et services sur mesure. Logiciels sur mesure à partir de composants standardisés.</i> Exemple caractéristique : <i>Système de gestion d'entreprise</i> Producteurs principaux : <i>S.S.I.I., sociétés de conseil, liées à des producteurs de matériel ou de progiciel</i> Utilisateurs principaux : <i>Grandes et moyennes entreprises, Administrations</i> Productivité dans la production des logiciels : <i>Assez élevée</i> Fiabilité des logiciels produits : <i>Assez élevée</i> Adéquation aux besoins des utilisateurs : <i>Dépend de la qualité de la relation de service</i>
Produits génériques	Monde de la création	Monde fordiste
	Type de produit : <i>Logiciels libres (code source)</i> Exemple caractéristique : <i>Le système d'exploitation Linux</i> Producteurs principaux : <i>Universités, centres de recherche, créateurs indépendants</i> Utilisateurs principaux : <i>Au départ communauté informatique</i> Productivité dans la production des logiciels : <i>Variable</i> Fiabilité des logiciels produits : <i>Amélioration rapide possible (pour les logiciels qui connaissent un succès initial)</i> Adéquation aux besoins des utilisateurs : <i>Forte pour la communauté informatique. Plus problématique pour les simples usagers.</i>	Type de produit : <i>Progiciels commerciaux (code objet et services limités)</i> Exemple caractéristique : <i>Tableur</i> Producteurs principaux : <i>"Editeurs" de logiciels</i> Utilisateurs principaux : <i>Ménages, entreprises (y compris P.M.E.), administrations</i> Productivité dans la production des logiciels : <i>Élevée</i> Fiabilité des logiciels produits : <i>Souvent insuffisante</i> Adéquation aux besoins des utilisateurs : <i>Uniquement pour des besoins standards</i>

A - LE MONDE INTERPERSONNEL DES LOGICIELS SUR-MESURE

Le monde interpersonnel correspond à la production *ex-nihilo* d'un logiciel sur mesure, développé en fonction des besoins spécifiques du client-utilisateur. Les principaux producteurs de ce type de logiciel sont les services informatiques internes de l'entreprise ou de l'administration utilisatrices ou, quand l'activité est externalisée, des S.S.I.I. le plus souvent spécialisées dans un domaine particulier. La quasi-absence d'économies d'échelle implique a priori une taille limitée des unités de production. Toutefois l'existence de projets d'une très grande envergure, et surtout l'importance des phénomènes de réputation dans le choix d'un producteur de logiciels en raison de l'incertitude forte de l'utilisateur sur les qualités du produit futur, peuvent entraîner la présence de producteurs d'une taille importante.

Pour la production qui est fournie par des prestataires externes, Gérard Dréan (1996 A) distingue deux types d'activités : d'une part, une *activité d'assistance technique* où chaque employé est détaché au près d'un client et intégré à une de ces équipes. Chaque individu est facturé indépendamment des autres et constitue donc l'unité de base de l'activité. Ce type de prestation peut être fourni par des informaticiens indépendants ou par des entreprises plus importantes, mais qui fonctionnent de fait comme un conglomérat de petites unités autonomes : pour l'essentiel chacun gère ses propres activités et seules quelques tâches administratives et commerciales sont centralisées au niveau de l'encadrement. D'autre part, une *activité de développement ou d'intégration* qui est constituée de projets réalisés par une équipe composée selon les besoins spécifiques du projet et qui peut varier au cours du temps. En général, ces prestations sont assurées par des entreprises de taille plus importante avec des structures plus complexes en raison d'une activité commerciale plus nettement séparée de l'activité de réalisation, et de l'existence possible de services de support centralisés pour répondre à des besoins de compétences critiques et de savoir-faire particuliers.

Dans les deux cas, les entreprises sont des entreprises de services à forte composante humaine, où la plupart des collaborateurs interviennent dans la relation avec le client et doivent prendre des initiatives autonomes (travail fréquent hors du cadre de l'entreprise, absence de supervision étroite) qui façonnent la réputation de l'entreprise.

A l'ère de l'informatique "traditionnelle" (cf. chapitre II), la production sur mesure des logiciels était la forme dominante d'une activité essentiellement destinée aux grandes entreprises et aux administrations. Même si elle conserve une place importante dans certains

domaines, cette forme de production décroît inexorablement en raison d'un positionnement particulièrement peu favorable par rapport aux trois dimensions critiques de la production des logiciels.

La première difficulté concerne la capacité du logiciel à résoudre de façon satisfaisante le "problème" de l'utilisateur. Une réponse positive à cette question suppose une compréhension réciproque des attentes et de la situation concrète des utilisateurs d'une part, des compétences et des possibilités des producteurs d'autre part. Elle nécessite une grande qualité de la communication entre utilisateurs et producteurs, qui implique une certaine stabilité des relations et la volonté d'y consacrer des ressources suffisantes. Or le développement des logiciels s'est effectué, surtout initialement, dans un contexte marqué par une domination forte des producteurs sur les utilisateurs : celle-ci s'appuyait à l'intérieur des organisations sur le pouvoir des informaticiens, qui détenaient des compétences techniques stratégiques ; sur le plan externe, la forte croissance de l'activité s'opérait sur un marché durablement déséquilibré en faveur de l'offre³⁹. Dans ces circonstances, les pressions sur les coûts et les délais aboutissaient à livrer des produits peu satisfaisants, d'autant plus que la productivité (pour la production des logiciels sur mesure) reste marquée par un niveau peu élevé et une faible croissance. Ceci s'explique par l'importance du travail pour mettre au point une application de ce type, qui est considérée comme correspondant à un problème original à résoudre et donc sans grande possibilité d'utiliser des développements déjà effectués. Enfin, la fiabilité du logiciel implique des tests exhaustifs de tous les cas possibles pour l'ensemble du logiciel développé spécifiquement qui alourdissent fortement les coûts et les délais de production. De ce fait la fiabilité de ce type de logiciel est en général limitée, ou quand il s'agit d'une dimension critique du logiciel considéré (espace, armement, banque...), les efforts pour augmenter la fiabilité dégradent encore la productivité.

Le monde interpersonnel de production des logiciels constitue ainsi la caricature du "cercle vicieux" de l'économie du logiciel, analysé dans le chapitre IV. Les qualités

³⁹ Gérard Dréan, qui a dirigé d'importantes sociétés de services informatiques et occupé des fonctions dirigeantes dans leur organisme professionnel, dresse un constat accablant des pratiques existantes : pour les prestations d'assistance technique où l'objectif est de maximiser le temps facturé passé chez le client, "à la limite peu importe que le travail soit inutile, de mauvaise qualité ou médiocrement productif tant que le client paie de bonne grâce" (1996 A, p. 276) ; pour l'activité de développement, la nécessité de "promettre beaucoup pour le prix le plus faible possible" fait que "d'une certaine façon, les qualités qui sont nécessaires pour réussir dans l'exécution d'un projet (lucidité, prudence, honnêteté,...) sont autant de raisons potentielles de ne pas emporter les contrats" (idem, p. 277).

insuffisantes (fiabilité, inadaptation) des logiciels génèrent une maintenance importante, qui influe négativement sur la productivité globale du secteur, qui à son tour maintient les producteurs de logiciels dans une position de force par rapport aux utilisateurs et permet d'imposer des logiciels très imparfaits.

B - LE MONDE FORDISTE DES PROGICIELS COMMERCIAUX

1 - L'essor des progiciels

Le monde *fordiste* correspond à la production de progiciels, commercialisés le plus souvent par ce qui est appelé traditionnellement des "éditeurs" de logiciels, expression peu satisfaisante pour décrire une activité qui en général englobe principalement la création des progiciels. Le terme de progiciel est la contraction de produit (au sens restrictif du terme, opposé à la notion de service) et de logiciel ; les progiciels sont des logiciels commercialisés prêts à l'emploi, utilisables indépendamment de leur auteur, définis pour un marché anonyme et non pour un utilisateur particulier. La production de progiciels est, économiquement parlant, radicalement différente d'une activité de services, et plus proche par sa logique industrielle de la production des composants informatiques dont elle partage de nombreux attributs (Gérard Dréan, 1996 A, p. 206).

Le progiciel étant défini unilatéralement par l'offreur dans l'espoir d'un marché, l'utilisateur n'a pas d'influence sur le contenu du produit, et doit éventuellement adapter ses besoins aux produits existants : pour les progiciels d'entreprises, Eurostaf estime que le choix d'un progiciel par rapport à un logiciel sur mesure est valide quand il permet une adéquation aux besoins à 80 %, en tenant compte des efforts éventuellement nécessaires de réorganisation de l'entreprise et de ses procédures pour pouvoir utiliser le progiciel (1996, p. 68). C'est qu'en effet un progiciel présente de nombreux avantages : rapidité de réalisation⁴⁰, évolutivité de la solution grâce à l'existence de nouvelles versions et surtout coût beaucoup plus faible en raison des économies d'échelle. Une étude de Bossard Consultant estime que l'échelle "durée-coût" varie de 1 à 8 entre un progiciel et un développement interne pour les banques de taille moyenne (01 Informatique, 13-10-1995). Cet écart explique que, si dans les années cinquante

⁴⁰ Le progiciel lui-même étant déjà disponible, le temps de réalisation comprend uniquement l'installation et l'adaptation au système informatique du client (paramétrage, interfaçage...), qui toutefois peut prendre un certain temps (cf. D).

et soixante les utilisateurs refusaient les progiciels pour les opérations comme la paie, la comptabilité, l'inventaire en estimant que leurs besoins étaient trop spécifiques, la situation s'est inversée radicalement dans les années quatre vingt, notamment quand le coût des logiciels a été rapporté au coût du matériel en forte baisse (Frederick P. Brooks, 1996, p. 171). C'est logiquement dans la micro-informatique où la baisse du prix des matériels a été la plus rapide, que l'utilisation des progiciels s'est généralisée, enclenchant un cercle vertueux de croissance de ventes d'un matériel rendu plus attractif par la disponibilité immédiate de progiciels divers et variés, et d'incitations à l'écriture de nouveaux logiciels pour un marché de plus en plus large. C'est ce qui va permettre d'étendre l'utilisation des logiciels à une proportion croissante de simples "usagers" (*end-user*) dont l'activité principale est extérieure au champ technologique de l'informatique (Marie de Besses, 1999, p. 271-273). De même, David C. Mowery souligne que ce sont les utilisateurs les moins "sophistiqués" qui emploient la plus forte proportion de progiciels (1996, p. 308). En conséquence, aussi bien sur le plan économique (produit déjà existant) que technologique (compétences limitées), les utilisateurs ont une influence réduite sur les progiciels. Leur seule possibilité d'action est un *exit*, qui consiste à effectuer un choix différent lors du renouvellement du progiciel. Toutefois cette opération n'est pas toujours possible, ni simple à réaliser (cf. infra).

Le constat de la forte croissance des progiciels, en général et sur longue période, a été effectué précédemment (cf. chapitre IV). La part des progiciels passe de 13,5 % du marché mondial des technologies de l'information en 1985 à 18,4 % en 1995 avec un taux de croissance annuel moyen de 16,3 % (OCDE, 1997 B, p. 16). Centrés au départ sur certains types de matériels (les micro-ordinateurs) et certains types de produits (les logiciels utilitaires généraux, notamment les systèmes d'exploitation), les progiciels s'étendent progressivement à l'ensemble de l'informatique et concernent de plus en plus d'applications. Par exemple, si en 1995 les progiciels représentaient 30 % des systèmes d'information de gestion (70 % pour les développements spécifiques) un organisme professionnel prévoyait que la proportion serait inversée cinq ans plus tard (CXP, 1995, p 1.14)⁴¹. Il est vrai que sur ce segment un certain nombre d'événements ont contribué à accélérer cette mutation en incitant au renouvellement des logiciels : ouverture des marchés et création de l' "espace unique", introduction de l'euro

⁴¹ Serge Bouchy effectue un constat identique : "si par le passé les applications dites "spécifiques" l'emportaient largement sur la mise en œuvre de progiciels généralisés, ces derniers gagnent chaque jour du terrain dans tous les domaines thématiques et sectoriels de l'entreprise" (1994, p. 313).

et surtout passage à l'an 2000. Cette forte croissance des progiciels, dont la poursuite est très généralement anticipée, se reflète dans les performances boursières particulièrement favorables de la plupart des entreprises de ce secteur (Robert Merges, 1996, p. 272, et Eurostaf, 1996 B, p. 30).

2 - Un fordisme spécifique

La production des progiciels peut être considérée comme relevant du monde fordiste, dans la mesure où elle intègre certaines caractéristiques décisives de ce monde de production. Il s'agit d'une production de masse de produits standardisés destinés à un marché anonyme. A la base de cette production de masse, se trouvent de forts gains de productivité dus à l'importance des économies d'échelle (cf. chapitre IV). De façon moins déterminante, la gestion d'un produit vendu massivement et donc relativement général - au sens où il offre une grande variété de fonctions en se pliant à des modalités d'utilisation diversifiées dans des configurations nombreuses - exige les organisations et les méthodes que l'on retrouve habituellement dans les entreprises commercialisant des produits industriels de masse. Il en est de même, pour le cycle de production qui est relativement long depuis les études de marché jusqu'aux tests du produit final et à la gestion des différentes versions du produit.

Mais par contre, sur certains points, les producteurs de progiciels ont des caractéristiques très différentes des entreprises fordistes habituelles, en raison des spécificités technico-économiques des progiciels, ce qui justifie de parler d'un "fordisme spécifique" dans le cas des progiciels. Ces spécificités concernent la production des progiciels, avec des coûts de production qui sont constitués presque exclusivement des coûts de développement de l'exemplaire original, et la diffusion des progiciels fortement influencée par l'existence de rendements croissants d'adoption particulièrement importants.

La nécessité d'un développement rapide

Nous avons vu que le secteur des progiciels se caractérisait de façon générale par la domination d'une entreprise sur chaque segment (cf. chapitre VI). Les effets de *feed-back*, positifs (renforcement des forts) et négatifs (affaiblissement des faibles), qui conduisent à cette situation, sont extrêmement rapides en raison de la vigueur des rendements croissants d'adoption, que ce soit l'apprentissage par l'usage (particulièrement élevés pour les progiciels), les interrelations technologiques (entre logiciels de natures différentes, système d'exploitation et applications par exemple), les économies d'échelle en production (potentiellement infinies

avec un coût de reproduction qui tend vers zéro), les rendements croissants d'informations (importance de la connaissance des progiciels existants⁴²) et les externalités de réseaux directes et indirectes (David C. Mowery, 1996, p. 10). Les externalités de réseaux directes (effet de club direct) proviennent du fait que la valeur d'un progiciel (un traitement de texte par exemple) dépend du nombre d'utilisateurs disposant du même progiciel, avec lesquels il va être possible de faire circuler des données formatées (des documents dans l'exemple du traitement de texte) avec ce progiciel⁴³. Les externalités de réseaux indirectes reposent sur le fait que la diversité de l'offre de produits complémentaires (extensions matérielles, autres applications "compatibles"...) dépend de la taille du réseau constitué par les possesseurs du progiciel. La particularité des externalités de réseaux, par rapport aux autres rendements croissants d'adoption, est que le comportement des autres utilisateurs a une influence sur la valeur du produit pour l'utilisateur après que celui-ci l'a acheté (et pas seulement avant l'achat). De ce fait, les externalités de réseaux confèrent aux anticipations une importance décisive (cf. chapitre II).

En conséquence, il est souvent déterminant pour une entreprise d'être la première présente sur un nouveau marché (Carl Shapiro, Hal R. Varian, 1999, p. 134). Ces auteurs considèrent qu'il s'agit de la forme la plus simple et la plus efficace des "tactiques de préemption" vis à vis du marché (idem, p. 239-240). L'avance initiale a tendance à s'amplifier rapidement et peut permettre de dominer le marché (Gérard Dréan, 1996 A, p. 137). Dans cette situation où la concurrence est vive sur les délais de réalisation (Frederick P. Brooks, 1996, p. 246), les entreprises qui s'imposent sont celles qui ont su adopter des méthodes de développement très rapides.

La production classique de logiciels avait été formalisée dans des méthodes de développement de complexité croissante, où les étapes de plus en plus nombreuses sont temporellement ordonnées (cf. chapitre III). Elle était essentiellement l'œuvre de grandes entreprises, dotées de formes de management et de cultures d'entreprise bien établies, ayant les caractéristiques du modèle bureaucratique et dont la meilleure illustration était le

⁴² Dans les technologies de l'information et de la communication, la difficulté à évaluer les coûts et les rendements réels des investissements pousse à adopter des comportements d'imitation (Alain Rallet, 1997, p. 86).

⁴³ La possibilité d'utiliser avec un progiciel des données produites sur un progiciel concurrent renvoie à des contraintes techniques mais surtout à des stratégies de verrouillage. Il est par exemple significatif que pendant longtemps le système d'exploitation des Mac (MacOS) pouvait lire les disquettes MSDOS alors que l'inverse n'était pas vrai (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 139).

fonctionnement d'IBM. Mais pour les progiciels, il ne s'agit pas, comme dans le cas des logiciels sur mesure, d'écrire un logiciel répondant à des spécifications précises définies contractuellement, mais d'aboutir le plus rapidement possible (en tout cas avant les concurrents) à un produit dont les caractéristiques répondent à des attentes encore peu formalisées et exprimées de futurs utilisateurs, susceptibles de constituer un nouveau marché. "L'industrie du progiciel (...) a vu le jour sous la forme de centaines d'entreprises nouvelles, qui se concentraient sur le travail à accomplir plutôt que sur le processus à suivre⁴⁴, et se sont inventées une culture" (Frederick P. Brooks, 1996, p. 246). L'exemple du fonctionnement de Microsoft⁴⁵, analysé par Michael Cusumano et Richard Selby (1996), est significatif : les caractéristiques précises du progiciel ne sont pas connues à l'avance ; il existe simplement un objectif assez général, affiné par des approximations successives⁴⁶ ; le développement est effectué en parallèle par des équipes de trente à quarante personnes responsables de leurs spécifications, de leurs délais et même du processus par lequel elles définissent, construisent et livrent l'ensemble d'options dont elles ont la charge (selon Jim McCarthy de Microsoft, cité par Frederick P. Brooks, 1996, p. 241). Ikujiro Nonaka et Hirotaka Takeuchi considèrent que Microsoft est une entreprise "conduite comme une entreprise japonaise : (...) les développeurs de software de Microsoft essaient de faire jaillir autant de connaissances tacites que possible au cours d'un processus rappelant l'approche de développement "sur le tas" ou "en mains" utilisée par les entreprises japonaises ; ils amènent le prototype sur le marché, proposent à des clients de l'essayer et enregistrent leurs avis et impressions sur la façon de l'améliorer ; ce processus d'essais et erreurs peut être répété à plusieurs reprises avant qu'un produit ne soit finalement introduit sur le marché" (1997, p. 266). Cette forme d'organisation permet une meilleure reconnaissance du talent du programmeur individuel (Frederick P. Brooks, 1996, p. 246) et une forte mobilisation des salariés⁴⁷.

⁴⁴ Maarten Boasson fait observer qu' "il est singulier que, presque toute l'industrie occidentale se conforme au modèle des processus, mais qu'en même temps elle se rende dépendante de Microsoft, fournisseur qui ne se préoccupe nullement de processus" (1998, p. 5).

⁴⁵ L'opposition entre la "culture IBM" et la "culture Microsoft" est apparue avec une particulière netteté quand elles ont tenté de développer ensemble le système d'exploitation OS/2. L'échec de cette coopération s'explique amplement par cette opposition (Paul Carroll, 1994).

⁴⁶ Chaque nuit une compilation complète du logiciel est effectuée permettant de juger ce que réalise (et ce que ne réalise pas !) le logiciel.

⁴⁷ Selon Michael Cusumano et Richard Selby, chez Microsoft il est courant que des programmeurs restent trois ou quatre jours d'affilée dans l'entreprise (1996, p. 96).

L'instabilité de l'environnement

Dans une activité où la plus grande part des coûts sont supportés en phase de développement, les entreprises qui arrivent à s'imposer rapidement sur un marché où elles peuvent se retrouver en situation de quasi-monopole, ont "des niveaux de profitabilité très élevés" (Gérard Dréan, 1996 A, p. 114)⁴⁸. Si le secteur du progiciel n'a pas connu globalement la dégradation de profitabilité du reste de l'industrie informatique à la fin des années quatre-vingt, il se caractérise toutefois par une plage très large des performances en terme de profit avec des maxima très élevés mais aussi des pertes très élevées pour certaines firmes (idem, p. 208-211). La dégradation des résultats d'une entreprise peut être extrêmement rapide (la force des effets *feed-back* joue également dans le sens négatif) dès qu'une entreprise se retrouve en porte à faux par rapport aux changements fréquents et difficilement prévisibles du secteur informatique⁴⁹. C'est là une deuxième différence avec le monde fordiste habituel caractérisé par "un environnement relativement simple et plutôt stable" (Jean Gadrey, 1994 A, p. 170). L'incertitude est très forte quand une entreprise lance un nouveau progiciel : "risque d'avoir mal perçu les besoins réels et prioritaires de la branche concernée, risque qu'un concurrent sorte plus vite un produit comparable, risque qu'un autre raccourcisse la vie prévue du produit par une amélioration astucieuse, risque de panne logicielle (bug), de mauvais fonctionnement, ou même risque d'un article mitigé dans la presse spécialisée" (Jean-Pierre Brulé, 1993, p. 40). L'objectif, qui est de déclencher un *feed-back* positif autour du nouveau produit, peut être atteint par des stratégies très différentes entre lesquelles il est délicat de choisir. C'est le cas en particulier par rapport aux technologies existantes de l'alternative entre incompatibilité ("pari de la performance") et compatibilité ("migration") étudiée par Carl Shapiro et Hal R. Varian (1999, p. 182). Enfin le succès peut être éphémère comme l'illustre le devenir de certaines entreprises de progiciels applicatifs pour micro-ordinateurs qui ont à un moment donné, dominé leur marché : Lotus, WordPerfect et Ashton Tate ont été rachetés, WordStar a fait faillite.

De ce fait, les entreprises qui arrivent à maintenir des résultats élevés sont celles qui ont des structures organisationnelles leur permettant de s'adapter vite aux évolutions imprévues.

⁴⁸ L'absence de limites de capacité de production et des coûts marginaux quasi-nuls conduisent par exemple Microsoft à avoir une marge brute de 92 % (Carl Shapiro, Hal R. Varian, 1999, p. 27).

⁴⁹ Par exemple, Novell qui avait le plus fort taux de profit du secteur des progiciels en 1991 et 1992 se retrouve déficitaire dès l'année suivante (Gérard Dréan, 1996 A, p.83-84).

C'est le cas de Microsoft, ce qui lui permet d'adopter rapidement des innovations incrémentales introduites par ses concurrents (Frédéric Dromby, 1999, p. 589), et ce qui lui a notamment permis d'effectuer un changement radical d'attitudes par rapport à Internet : après avoir complètement négligé ses potentialités, dont la meilleure illustration était que le service d'informations en ligne de Microsoft (MSN) n'était pas un site Internet, Microsoft va à partir de 1995 concentrer ses efforts sur la production de progiciels pour Internet (navigateur, serveurs...) et sur l'intégration de fonctionnalités pour Internet à l'ensemble de ses progiciels existants.

La spécificité de la production des progiciels

La troisième différence renvoie à la spécificité de la production des progiciels dont l'essentiel de l'activité réside dans la création de l'original. Ceci a comme conséquence que si le progiciel est en tant que tel un produit standardisé, sa production ne nécessite nullement une standardisation des tâches, à l'inverse de ce qui existe dans la production de masse des biens industriels et des services. Au contraire, le développement d'un progiciel original est une "création". Or, "la créativité provient des individus et non des structures ou des processus, (...), le problème principal [étant]de savoir comment concevoir les structures et les processus pour améliorer, plutôt qu'inhiber, la créativité et l'initiative" (Frederick P. Brooks, 1996, p. 240). Maarten Boasson explique qu'il est nécessaire de recourir à une "approche créative" opposée à une "approche mécanique" pour le développement des logiciels, systèmes complexes "qui ne peuvent être effectués (...) par des hommes pratiquement réduits à un rôle machinique" (1998, p. 7). Il s'agit "d'un processus de développement créatif et assez chaotique, dans lequel des décisions doivent être prises simultanément à de nombreux niveaux d'abstraction, [et qui] se prête mal à une approche classique du management" (idem)⁵⁰. La conception de logiciels nécessite une étroite coopération entre des types de spécialistes différents, notamment entre ceux qui possèdent des compétences du domaine concerné et ceux qui possèdent des compétences proprement informatiques : construire un logiciel "exige que l'on se communique des informations techniques très complexes afin que des décisions intelligentes puissent être prises" constate Lauren Ruth Wiener, qui estime que "les

⁵⁰ Selon Daniel Ichbiah, la force principale de Microsoft est qu'elle a réussi "à canaliser la créativité débridée de cette population fantasque" que représentent les programmeurs, et est "parvenue à résoudre une équation complexe : amener de purs artistes, indépendants et enclins à se perdre dans d'infimes détails dans leur recherche de la perfection pure, à coordonner leurs efforts et à mettre leur génie au service d'un produit destiné à un marché de masse" (1995, p. 389).

hiérarchies de domination (...) ne sont pas vraiment adaptées au développement de logiciels" (1994, p. 106-107).

Ces caractéristiques particulières, qui font de la production de progiciels un "fordisme spécifique", expliquent les performances contrastées de ce monde de production.

3 - Des performances contrastées

La production de progiciels permet une productivité beaucoup plus élevée que celle qui existe dans la production de logiciels sur mesure (cf. chapitre V). Certes le développement d'un progiciel destiné à des milliers, voire à des millions d'utilisateurs, demande davantage de travail que le développement d'un logiciel comparable destiné à un seul utilisateur, parce que le progiciel doit pouvoir être utilisé par des utilisateurs hétérogènes et dans des environnements matériels et logiciels très divers. Mais dans le cas du progiciel cet effort de développement sera divisé par le nombre de progiciels commercialisés, d'où une croissance de la productivité quasiment proportionnelle au succès du progiciel mesuré par le nombre d'exemplaires vendus, les coûts de production d'un exemplaire supplémentaire étant très faibles (voire nuls si le progiciel est téléchargé). De ce fait, l'introduction des progiciels a permis de faire baisser spectaculairement le prix des logiciels pour l'utilisateur.

Une qualité parfois insuffisante

Par contre, le bilan apparaît nettement moins positif en ce qui concerne la qualité des logiciels. Une première limite est inhérente à la nature même d'un progiciel. Etant produit en grande série, il ne peut par définition que répondre aux besoins standard de demandeurs anonymes. L'utilisation d'un progiciel ne sera donc pertinente que pour des traitements peu spécifiques (ce qui correspond à une grande partie des utilisations de l'ordinateur comme outil individuel, et à certaines utilisations au niveau d'une entreprise comme la comptabilité) ou pour des traitements où l'adaptation des spécificités de l'utilisateur (individu ou entreprise) au progiciel n'est pas jugée rédhibitoire⁵¹.

La deuxième limite s'explique par le processus, précédemment analysé, qui permet à un progiciel donné de s'imposer sur un marché. Si les méthodes de production les plus efficaces

⁵¹ En bonne logique économique il faut comparer le coût de l'adaptation au progiciel, à l'économie que représente l'utilisation d'un progiciel par rapport à un logiciel sur mesure.

des progiciels permettent de disposer à tout moment d'un produit commercialisable, le choix de la date où ce produit est lancé sur le marché dépend plus de considérations stratégiques (être présent sur le marché avant les concurrents) que de l'achèvement réel du produit. Ceci a des conséquences très négatives sur la fiabilité des premières versions des progiciels⁵². Alors que leur diffusion auprès de multiples utilisateurs devrait constituer une garantie de bon fonctionnement, il n'est pas rare que certains progiciels soient commercialisés avec un nombre anormalement élevé d'erreurs : "une arrivée précoce signifie parfois des compromis sur la qualité et des risques accrus de bugs" (Carl Shapiro, Hal R. Varian, 1999, p. 240). Par exemple, la version 3 de Word contenait plus de 700 bogues quand elle a été commercialisée en 1987 (Michael Cusumano, Richard Selby, 1996)⁵³, la première version vendue de Windows 3 (la version 3.0) a été considérée comme étant une version *beta*⁵⁴ officielle (Roberto Di Cosmo, Dominique Nora, 1998, p. 48). Plus franche mais assez étonnante est la récente décision prise par Apple de vendre la version *beta* de son nouveau système d'exploitation (Mac OS X), logiciel qui est au cœur de la stratégie de reconquête de l'entreprise et qui avait pris du retard dans son développement (18h.com, mercredi 13 septembre 2000, n° 368).

Le mécontentement légitime des utilisateurs n'a le plus souvent que peu de conséquences vues les faibles possibilités d'*exit*. En effet, les carences des logiciels n'apparaissent généralement qu'après une période d'utilisation qui peut être plus ou moins longue, alors qu'un progiciel peut réussir à s'imposer rapidement en raison des dynamiques d'adoption sur des marchés qui restent rarement longtemps concurrentiels et où les utilisateurs se retrouvent "verrouillés". Les producteurs sont en position de force face aux utilisateurs⁵⁵,

⁵² Il faut ajouter qu'un mode de développement du type de celui de Microsoft, où les spécifications définitives ne sont déterminées qu'à la fin, accélère le processus de développement mais rend plus difficile l'élimination des erreurs (Christian Genthon, Denis Phan, 1999, p. 179).

⁵³ Selon Lauren Ruth Wiener (1994) il fallut tout le génie de Bill Gates pour réussir à faire une démonstration publique de son utilisation sans que l'ordinateur ne se bloque.

⁵⁴ Une version *beta* est une version provisoire dont on sait qu'elle n'est pas encore au point et qui est incomplète au niveau des fonctionnalités.

⁵⁵ Cette situation existe même quand l'utilisateur est une grande entreprise, comme le montre l'exemple de Bell Atlantic, victime d'un "verrouillage" c'est à dire d'une situation où les coûts de changement de technologie sont importants. Bell Atlantic a équipé son réseau téléphonique d'un système de commutation numérique développé par ATT et qui fonctionne avec un système d'exploitation "propriétaire" d'ATT (Bell n' a pas le contrôle du code source). De ce fait, chaque fois que Bell souhaite faire une extension (numéros verts, numérotation vocale) ou effectuer une nouvelle connexion à différents périphériques, elle doit faire appel à ATT

ce qui se reflète dans l'absence d'obligation de résultat existant pour les progiciels. Les utilisateurs vont se retrouver contraints d'acheter les versions ultérieures des progiciels, qui corrigent les premières bogues constatées⁵⁶. Se trouvent vérifiées dans le cas des progiciels, les conclusions de Dominique Foray (1990), selon lequel c'est parce qu'une technologie est adoptée qu'elle devient plus efficace que des technologies concurrentes qui au départ pouvaient avoir plus de potentialités, et de Brian Arthur (1989) pour qui les rendements d'échelle croissants font que des entreprises peuvent parvenir à une position de quasi-monopole avec des technologies qui deviennent totalement incontournables bien qu'elles ne soient pas toujours les meilleures (cf. également chapitre II).

La plupart des spécialistes de ce secteur en concluent que la qualité d'un progiciel est un facteur secondaire par rapport à d'autres critères : le savoir-faire marketing, la puissance de la distribution et bien sûr la compatibilité avec les applications existantes (Roberto Di Cosmo, Dominique Nora, 1998, p.p. 79). Carl Shapiro et Hal R. Varian indiquent que "la conclusion est aussi nette qu'effrayante : le succès comme l'échec sont autant déterminés par les anticipations des consommateurs et la chance que par la valeur intrinsèque du produit" (1999, p. 163). De même Jean-Marie Desaintquentin et Bernard Sauter estiment que "la qualité intrinsèque d'un logiciel garantit beaucoup moins son succès que sa reconnaissance comme un standard, respecté par le plus grand nombre possible de professionnels, fournisseurs et utilisateurs" (1991, p. 191).

Les rendements croissants d'adoption font qu'un progiciel peut dominer durablement un marché même si ce progiciel n'est pas (ou plus) jugé optimal par une majorité d'utilisateurs. Toutefois le verrouillage des utilisateurs n'est jamais absolu. Les rendements croissants d'adoption sont différents selon le type d'informatique -ils sont plus importants dans

pour le développement d'interfaces ou la mise à jour nécessaire du système d'exploitation. ATT est évidemment en position de force notamment sur le plan financier : selon Bell Atlantic, le marché des mises à jour logicielles des systèmes d'ATT représente 40 % du chiffre d'affaires réalisé par cette entreprise dans le secteur de la commutation (Carl Shapiro, Hal R. Varian, 1999, p. 100).

⁵⁶ Il faut toutefois noter que si les versions ultérieures peuvent apporter des améliorations significatives, elles sont pour certains aspects qui peuvent être importants, dépendantes des choix initiaux qui déterminent les possibilités de "migration" du progiciel. Un exemple célèbre est MSDOS qui avait été conçu pour gérer une mémoire vive de 640 KO, à une époque où la mémoire vive courante d'un micro-ordinateur était de 64 KO et surtout parce que IBM ne voulait pas faire développer un système d'exploitation qui aurait pu être utilisé sur des machines plus puissantes comme ses mini-ordinateurs. Cette contrainte pèse encore maintenant sur les dernières versions de Windows – même si à chaque nouvelle version de Windows, Microsoft assure qu'il va s'en libérer. Pour être contournée et exploiter au mieux les quantités de mémoire vive bien supérieures actuellement disponibles, il a fallu des trésors d'astuces de la part des programmeurs qui ont certes réussi à améliorer les performances des micro-ordinateurs mais non sans des conséquences négatives sur la fiabilité du système.

l'informatique traditionnelle que dans la micro-informatique - et selon le type de progiciels – ils sont plus importants pour les systèmes d'exploitation que pour les progiciels applicatifs. Concernant le type d'applications, les situations sont également très différentes : un exemple de verrouillage important est le cas d'une application de *workflow* réalisée dans l'environnement Lotus Notes ; celle-ci sera très coûteuse à porter dans un autre environnement en raison de l'utilisation du langage Lotus Script utilisé pour définir les flux d'informations et qui n'est disponible que dans l'environnement Lotus ; un changement d'environnement nécessite donc de redéfinir les flux d'informations dans un autre langage (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 109). A l'opposé le changement de navigateur sur Internet s'effectue sans trop de difficultés (l'apprentissage est rapide et les données utilisées comme les fichiers de signets se transfèrent relativement facilement). Les principales applications bureautiques se situent entre ces deux extrêmes; avec des verrouillages d'intensité croissante pour le traitement de texte, le tableur et le système de gestion de bases de données.

Des améliorations qualitatives par changement d'entreprises leaders dans les logiciels d'application pour micro-ordinateurs

Concernant les logiciels d'application pour micro-ordinateurs, Frédéric Dromby (1999) montre que si pour chaque produit une entreprise leader a toujours largement dominé le marché, celle-ci n'a pas toujours été la même :

Tableau LII
Les entreprises leaders sur les marchés des logiciels d'application pour micro-ordinateurs (avec leur part de marché)

Type de logiciel	Première génération	Deuxième génération	Troisième génération
Traitement de texte	WordStar (65 % en 1982)	WordPerfect (67 % en 1988)	Word (55 % en 1996)
Tableur	VisiCalc (100 % en 1979)	Lotus 1-2-3 (68 % en 1988)	Excel (66 % en 1994)
Base de données	DBase III (63 % en 1988)	Paradox (45 % en 1991)* DBase IV (35 %)*	Access (34 % en 1994)

* Les deux produits appartiennent à Borland, celui-ci ayant racheté Ashton-Tate producteur de dBase en 1991.

Source : Frédéric Dromby (1999, p. 264)

L'explication de la succession d'entreprises différentes qui dominent chaque marché, avancée par Frédéric Dromby repose sur les notions d'*exit* et de *voice* de Albert O. Hirschman (1972) et sur une distinction entre "clients innovateurs" et "clients fidèles" ou "compatibilistes" (Frédéric Dromby, 1999, p. 342). Une entreprise introduisant des innovations qualitatives significatives peut s'emparer du marché, en raison de "l'excès d'inertie de la firme dominante" : celle-ci confrontée initialement à un *exit* limité et progressif de ses clients les plus exigeants au profit de la nouvelle entreprise innovatrice, subit en conséquence une *voice* moins importante de ses utilisateurs sur des marchés en forte croissance, ce qui ne l'incite pas à réagir – elle reçoit une "information trompeuse involontaire" de la part du marché ou "effet d'occultation" (idem, p. 372) - jusqu'à ce que le nouveau produit atteigne une masse critique qui fait basculer brutalement l'essentiel du marché. Chaque changement d'entreprise leader sur un marché correspond donc à une amélioration qualitative significative du type de progiciel correspondant.

Les facteurs de la puissance de Microsoft

Si cette thèse, limitée aux progiciels d'application pour micro-ordinateurs, nous semble convaincante pour le passage de la première à la deuxième génération de produits, pour lequel l'introduction d'innovations réelles est incontestable, il n'en est pas de même pour les produits de la troisième génération. A la différence des générations précédentes où les progiciels dominants étaient produits par des entreprises différentes, c'est la même entreprise (Microsoft) qui domine les trois marchés⁵⁷ avec des parts de marché qui vont d'ailleurs continuer à croître fortement durant les années quatre vingt dix : dès 1997, Microsoft détient 87 % du marché des suites bureautiques, qui deviennent la forme dominante de distribution de ces produits (Frédéric Dromby, 1999, p. 231). Contrairement à ce qu'affirme Frédéric Dromby, la place acquise par Microsoft repose moins sur une supériorité qualitative de ses produits⁵⁸ que sur le

⁵⁷ Nous avons négligé un quatrième marché, étudié par Frédéric Dromby, les logiciels de présentation, qui est également très largement dominé pour la troisième génération de produits par Microsoft avec PowerPoint.

⁵⁸ Frédéric Dromby utilise deux moyens pour juger du degré d'innovation introduit par un progiciel : d'une part, une enquête par questionnaire auprès d'experts, dont il reconnaît qu'elle n'est pas statistiquement valide (1999, p. 726) et dont les résultats concernant les produits de la troisième génération ne confirment pas un degré d'innovation important ; d'autre part, les jugements d'une presse "spécialisée" dont l'objectivité peut être mise en doute quand on sait que leur principale source de revenus est la publicité, dont la part la plus importante est fournie par ... Microsoft. On peut légitimement leur opposer de nombreuses études techniques effectuées par des chercheurs en informatique : par exemple Roberto Di Cosmo (1998) juge ces produits "médiocres" et estime que "au départ, les produits Microsoft étaient très nettement inférieurs à ceux de la concurrence" (p. 33 et 36).

pouvoir, financier et technologique, que lui conférait sa position monopolistique acquise sur les systèmes d'exploitation pour micro-ordinateurs. La puissance financière de Microsoft (dès le milieu de 1991, le chiffre d'affaires de Microsoft, constitué pour la plus grande part de la vente des systèmes d'exploitation, est supérieur à la somme de ces quatre suivants avec des taux de profit bien supérieurs) lui permet de s'imposer sur certains marchés où elle n'était pas présente en rachetant des entreprises (PowerPoint pour les logiciels de présentation en 1987, FoxPro pour les systèmes de gestion de bases de données en 1992) et en commercialisant à bas prix ses progiciels (Access développé à partir de FoxPro est vendu 99 dollars quand les SGBD se vendent en général au-dessus de 600 dollars). Sur le plan technologique, Microsoft en maîtrisant l'évolution du système d'exploitation, peut retarder délibérément la diffusion des API (*Application Programming Interface*) dont la connaissance est indispensable pour écrire les applications correspondantes, en argumentant qu'elle veut s'assurer de leur fonctionnement correct. C'est ce qui permettra à ses propres équipes de développement de fournir des versions des progiciels applicatifs adaptées aux différentes versions de Windows bien avant ses concurrents (Carl Shapiro, Hal R. Varian, 1999, p. 179)⁵⁹.

Mais surtout contrairement aux prévisions de Frédéric Dromby, qui estime que la quatrième génération de progiciels applicatifs, correspondant à une "informatique distribuée" liée aux développements des réseaux, sera dominée par IBM avec Lotus, c'est Microsoft qui renforce sa domination sur l'ensemble de la micro-informatique, voire qui l'étend à l'ensemble de l'informatique, en utilisant des méthodes qui ont peu à voir avec la qualité des produits.

Quand Microsoft en 1995 prend conscience de l'importance d'Internet, elle tente de s'imposer sur le marché des navigateurs, jugeant, certainement à tort, que ce segment allait occuper un rôle stratégique déterminant de même nature que les systèmes d'exploitation pour la micro-informatique. Ce marché, au départ complètement dominé par Netscape, va être conquis en quelques années par Microsoft. Microsoft commence par racheter la licence d'un navigateur (Mosaic) à Spyglass à partir duquel elle va développer Internet Explorer⁶⁰. Elle va

⁵⁹ Roberto Di Cosmo et Dominique Nora vont jusqu'à affirmer que Microsoft utilise sa maîtrise du système d'exploitation pour "saboter les produits concurrents" de ses progiciels applicatifs par des pratiques complexes et difficiles à prouver (1998, p. 93-96).

⁶⁰ Il est intéressant de noter que le logiciel de navigation de Netscape est également issu de Mosaic. La création de Mosaic, qui facilita l'accès du Web à un public important et son développement, fut effectuée sous forme de logiciel libre à l'université d'Urbana Champaign (Illinois). Un des auteurs, Marc Andreessen, étudiant dans cette université, va fonder avec Jim Clark, ex-PDG de Silicon Graphics, Mosaic Communications en avril

réussir à imposer Internet Explorer, pourtant jugé largement inférieur, au moins au départ, au produit de Netscape (Carl Shapiro, Hal R. Varian, 1999, p. 253), en agissant sur la clientèle "influente"⁶¹, en passant des accords avec les fournisseurs d'accès à Internet dont le rôle est souvent déterminant dans le choix du navigateur par un nouvel utilisateur, voire en payant les fournisseurs d'accès et les constructeurs de micro-ordinateurs (idem, p.256), en négociant des accords avec les développeurs de sites pour qu'ils ajoutent des caractéristiques qui ne fonctionnent qu'avec Internet Explorer⁶², puis en fournissant gratuitement Internet Explorer⁶³ et enfin en l'intégrant purement et simplement à Windows 98. C'est cette pratique radicale (tout acheteur de Windows se retrouve avec Internet Explorer qui est même très difficile à désinstaller) décisive dans le succès d'Internet Explorer, qui va susciter le procès de Microsoft⁶⁴.

Si l'affrontement avec Netscape a été le plus spectaculaire, il ne constitue qu'un épisode des efforts de Microsoft pour s'imposer dans de multiples directions (cf. chapitre VI). Le développement de Microsoft repose moins sur des innovations significatives qu'elle aurait introduite⁶⁵ que sur l'achat et l'adaptation (voire parfois la copie pure et simple) de technologies existantes (comme pour le DOS, Internet Explorer, Windows NT) dans certains cas avec pour seul objectif de "tuer" la technologie ou le produit acheté en suivant une

1994 pour diffuser un programme issu de Mosaic mais sécurisé et adapté au grand public, rebaptisé Netscape en novembre 1994.

⁶¹ Par exemple, Microsoft a signé un accord avec Dow Jones pour que les utilisateurs d'Internet Explorer accèdent gratuitement au Wall Street Journal (Carl Shapiro, Hal R. Varian, 1999, p. 215).

⁶² Les différentes versions d'Internet Explorer ne sont pas conformes aux standards définis par le W3C alors que Microsoft participe à l'élaboration de ces standards (cf. chapitre II).

⁶³ Ce qui a comme conséquence que Microsoft ne verse pas de redevance à Spyglass puisque Internet Explorer est offert (Roberto Di Cosmo, Dominique Nora, 1998, p.p. 107).

⁶⁴ Beaucoup d'observateurs craignaient que grâce à Internet Explorer, Microsoft puisse contrôler le commerce sur Internet, crainte avivée par la déclaration de Nathan Myrhvold, dirigeant de Microsoft, qui avait indiqué que l'objectif de Microsoft était de tirer profit de chaque transaction sur Internet (Carl Shapiro, Hal R. Varian, 1999, p. 257). Même si pour le moment Microsoft a perdu ce procès, et est dans l'attente des sanctions, ceci ne l'empêche pas de continuer à recourir à ce type de pratique : dans la dernière version de Windows, Windows ME sortie en septembre 2000, sont inclus Internet Explorer, mais aussi Windows Media 7 qui est en concurrence avec RealPlayer.

⁶⁵ Jusqu'en 1995, Microsoft n'avait aucune activité de recherche (Roberto Di Cosmo, Dominique Nora, 1998, p.76).

stratégie de *buy out* (Roberto Di Cosmo, Dominique Nora, 1998, p. 75 et 108)⁶⁶. La stratégie de Microsoft a été résumée dans la formule "*étendre et s'étendre*", qui consiste à racheter toute entreprise possédant une application susceptible de connaître le succès, la seule limite étant un éventuel veto du *Department of Justice* comme dans le cas de Intuit. Si les débuts de la micro-informatique avaient été marqués par les succès spectaculaires de certaines des nombreuses nouvelles entreprises innovantes qui avaient vu le jour dans un secteur où les barrières à l'entrée étaient faibles, la situation est maintenant bien différente, au point que selon Carl Shapiro et Hal R. Varian, la stratégie de beaucoup de jeunes entreprises est : "fabriquer un produit nouveau, capturer le marché correspondant et se faire racheter par Microsoft" (Carl Shapiro, Hal R. Varian, 1999, p. 245). Une indication complémentaire de la puissance acquise par Microsoft est fournie par le fait que dans certaines administrations aux Etats-Unis, l'achat d'un logiciel Microsoft ne nécessite pas de justificatif de la pertinence de son choix par celui qui passe la commande alors qu'une justification est nécessaire pour un logiciel d'une autre société (Antoine Brisset, 1999).

La situation de verrouillage technologique de Microsoft, qui devient de moins en moins réversible au fur et à mesure qu'elle s'étend sur une part croissante de l'économie du logiciel, peut être jugée préoccupante pour les qualités des futurs produits, alors même qu'elle s'appuie également sur des pratiques complémentaires qui peuvent apparaître très contestables. Précisons que ces pratiques discutables ne sont pas l'apanage du seul Microsoft, et que leur plus grande fréquence dans ce secteur que dans le reste de l'économie s'explique par les caractéristiques de ces marchés - de type *Winner Take All* - et les spécificités de la production - avec la part très importante des investissements irréversibles (même s'ils sont essentiellement immatériels) - qui peuvent expliquer (mais non justifier) des comportements à la limite de la légalité⁶⁷.

⁶⁶ Il faut reconnaître que cette pratique n'a pas été inventée par Microsoft : déjà en 1985, Lotus avait racheté les droits sur VisiCalc pour en stopper la diffusion (750 000 exemplaires avaient été vendus) au profit de son tableur 1-2-3 (Frédéric Dromby, 1999, p. 558).

⁶⁷ L'illégalité de certaines de ces pratiques n'est pas simple à établir. Si l'on considère la législation américaine sur la concurrence, le Sherman Act (1890) indique qu'il est illégal de monopoliser un marché, mais le plus souvent seul est sanctionné l'abus de la position de monopole, non la position de monopole elle-même. Toutefois les monopoles obtenus par le jeu d'acquisitions, de politiques de prix prédatrices, d'accords d'exclusivité, de ventes liées ou en packages peuvent être contestés. Enfin un monopole ne peut pas employer certaines stratégies alors que ces dernières sont largement répandues et parfaitement légales dans un environnement concurrentiel, comme par exemple, apporter à ses produits des modifications qui ont pour effet d'étendre le pouvoir de monopole ou de réduire l'éventail de choix des consommateurs (Carl Shapiro, Hal R. Varian, 1999, p. 264-267).

La première de ces pratiques concerne la *manipulation des anticipations des utilisateurs*. En présence d'externalités de réseaux, les anticipations constituent "un facteur clé de la décision des consommateurs concernant l'adoption d'une nouvelle technologie" (Carl Shapiro, Hal R. Varian, 1999, p. 237) et de multiples moyens sont utilisés pour convaincre les utilisateurs du futur succès d'un progiciel. On peut citer les 200 millions de dollars en communication dépensés par Microsoft pour le lancement de Windows 95 (OCDE, 1997 B, p. 69), ou les multiples pages de publicité placées dans les journaux et magazines par Sun en 1987, avec la liste des entreprises qui soutenaient Java pour convaincre que ce langage allait s'imposer (Carl Shapiro, Hal R. Varian, 1999, p. 20)⁶⁸. Mais une autre pratique s'est également largement répandue dans l'informatique, au point qu'une expression a été inventée pour la désigner : le *vaporware*. Il s'agit de l'annonce d'un produit qui ne voit pas le jour, ou à une date très éloignée de la date indiquée, afin de geler les ventes des concurrents. Cette pratique, initiée pour le matériel dans les années soixante-dix par IBM, a été suivie dans les progiciels par Lotus sur le marché des tableurs et par Ashton-Tate sur le marché des logiciels de gestion de bases de données⁶⁹, et par Microsoft. Cette pratique, qui peut être très efficace est difficile à prouver : comment faire la part entre un mensonge délibéré, et une sortie retardée pour des raisons techniques ?

La deuxième de ces pratiques est la *vente "forcée"* de progiciels avec d'autres progiciels ou avec du matériel. Microsoft a eu largement recours à ce type de pratique pour les systèmes d'exploitation (il était quasiment impossible jusqu'à une date récente d'acheter un micro-ordinateur sans Windows), qu'elle avait appuyé sur une politique assez spéciale de tarification aux constructeurs, à savoir une facturation des systèmes d'exploitation en fonction du nombre de micro-ordinateurs vendus, quel que soit le système d'exploitation qui leur est éventuellement adjoint (cf. chapitre III).

La troisième de ces pratiques consiste à contraindre les utilisateurs à acheter les *versions successives* des progiciels. La faiblesse des coûts de fabrication (au sens restreint, hors les coûts de conception), y compris pour reproduire un produit modifié, permet la mise sur le marché à intervalles très rapprochés de différentes versions d'un progiciel. Par contre un

⁶⁸ L'importance des effets d'annonce peut être illustrée par la plainte de Word-Perfect contre Microsoft qui avait annoncé que Word était le logiciel de traitement de texte le plus populaire du monde.

⁶⁹ A un point tel qu'une plaisanterie courante était que ces deux entreprises allaient fusionner sous le nom de *LATE*.

progiciel ne "s'usant" pas, son obsolescence ne peut provenir que de l'existence d'un produit nouveau perçu comme lui étant supérieur. Or pour la majorité des utilisateurs les fonctionnalités introduites dans les nouvelles versions présentent souvent un intérêt limité et ne justifieraient pas l'achat d'une nouvelle version. Microsoft arrive néanmoins à faire adopter les versions successives de ses progiciels, en utilisant les formats des fichiers produits avec les différentes applications. Ces formats, qui ne sont pas publiés et que les autres producteurs ne peuvent utiliser, évoluent de telle sorte qu'il est difficile voire impossible de lire un fichier produit par une application sur la même application mais d'une version antérieure⁷⁰. La nécessité de faire circuler facilement les fichiers entre les utilisateurs les conduit à acheter les dernières versions des différents progiciels.

Les formats de fichier constituent un exemple des pouvoirs de marché qu'offrent la maîtrise de standards dans leur évolution comme dans leur périmètre d'application (leur fermeture aux applications concurrentes). Cette stratégie est sciemment développée par Microsoft, notamment face à la menace que pourrait représenter les logiciels libres, comme le prouve un mémorandum interne à Microsoft⁷¹ : "En étendant les fonctions de nos protocoles et en en développant de nouveaux, nous pouvons interdire aux projets de logiciels libres l'accès au marché".

Un bon exemple en est l'attitude de Microsoft par rapport à Java développé par Sun. La perspective d'un langage universel, permettant à une application écrite dans ce langage de s'exécuter quel que soit le matériel utilisé, est contraire aux intérêts de Microsoft. Selon une attitude assez classique, Microsoft a affirmé soutenir Java⁷², tout en lui apportant des "améliorations" et en ajoutant des méthodes supplémentaires, qui ne fonctionnent que dans l'environnement Windows, ces modifications étant de fait destinées à contrer Java en limitant son champ d'application (Carl Shapiro, Hal R. Varian, 1999, p. 141 et 219). Cette

⁷⁰ Pour certaines applications (traitement de texte) la conversion est possible à condition de ne pas utiliser les dernières fonctionnalités, mais n'est pas toujours évidente à réaliser sans altération du fichier. Mais il est tout simplement impossible d'utiliser une base de données effectuée avec Access 97, sur une version précédente d'Access.

⁷¹ Appelé Mémorandum d'Halloween en raison du jour où il a été divulgué, son authenticité a été reconnue par Microsoft (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 72 et 86).

⁷² Pour assurer le développement de Java, Sun a concédé la licence de Java au plus grand nombre de sociétés, y compris à Microsoft, son plus farouche adversaire. Microsoft a habilement négocié le droit d'"améliorer" Java dans le cadre de l'accord de licence.

"microsoftisation de Java" (Roberto Di Cosmo, Dominique Nora, 1998, p. 113) fait d'ailleurs l'objet d'un procès de la part de Sun.

Ces analyses permettent de comprendre pourquoi si les logiciels constituent une excellente réponse au problème de la productivité dans la production des logiciels, le bilan est beaucoup moins favorable pour les questions ayant trait à la qualité des logiciels utilisés. Ces insuffisances sont une des raisons du développement de logiciels libres dans le cadre du "monde de la création".

C - LE MONDE DE LA CREATION DES LOGICIELS LIBRES

Dans l'économie du logiciel, le monde de la création correspond au développement des logiciels libres, qui occupent une place à part, mais non négligeable, même si, en raison de leur gratuité, leur importance est souvent statistiquement difficile à mesurer. Ce monde est le lieu de la création originale de logiciels très divers (depuis des systèmes d'exploitation ou des compilateurs jusqu'à de petits "utilitaires", en passant par les principaux outils logiciels nécessaires au fonctionnement d'Internet). Les deux succès les plus spectaculaires sont le système d'exploitation Linux qui est utilisé par plus de 25 % des serveurs Internet⁷³, et le logiciel serveur Web "Apache" qui domine son marché avec plus de 60 % de part de marché (Jean-Michel Dalle, Nicolas Jullien, 2000, p. 2). Le monde de la création englobe des producteurs très différents, qui peuvent parfois coopérer pour coproduire les logiciels. Ces producteurs peuvent être de simples particuliers (éventuellement regroupés en association ou liés par des relations informelles), des universités, des centres de recherche publics ou dépendants d'entreprises privées mais dont l'objectif principal n'est pas la rentabilité immédiate. L'exemple le plus célèbre d'un centre de recherche privé, dont on peut considérer qu'il faisait partie du monde de la création, est le Centre de recherches de Xerox à Palo Alto ou Xerox PARC. Ce centre avait été créé grâce à la santé financière de Xerox, liée à son avance technologique dans le domaine de la photocopie, avec la vocation de faire des recherches portant sur le long terme, tout en laissant la plus grande liberté à ses chercheurs et en ne leur assignant aucun objectif commercial (Jean-Yvon Birrien, 1990, p. 84). Si le succès

⁷³ D'après Dataquest, Linux a connu une croissance de 212 % en 1998 et est utilisé par 13 % des sociétés américaines.

technologique du PARC fut indéniable – "c'est dans ce lieu qu'on a initié la plupart des outils de la technologie informatique dont nous ferons usage jusqu'à la fin du siècle" (Robert X. Cringely, 1994, p.70) – Xerox n'exploitera pas les retombées commerciales de ces découvertes, et beaucoup de ses chercheurs partiront vers d'autres entreprises avec leurs inventions.

Selon certains adeptes des logiciels libres, les effets de leur développement peuvent être comparés à ceux de la micro-informatique, dans la possibilité de conduire à une réduction des coûts, à rendre certaines technologies plus accessibles et à susciter l'émergence de nouveaux services (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 175).

1 - L'originalité des logiciels libres

a - Un logiciel libre est un logiciel dont le code source est librement accessible...

Les logiciels libres sont des "logiciels disponibles sous forme de code source, librement redistribuables et modifiables" (extrait des statuts de l'Association Francophone des Utilisateurs de Linux et des Logiciels Libres). Ces logiciels ne sont pas dans le domaine public pour éviter qu'ils puissent être revendus sans fournir leur code source (cf. chapitre III). Pour protéger les logiciels contre tout verrouillage technique ou légal de leur utilisation, de leur diffusion et de leur modification, des licences publiques ont été créées en utilisant à rebours le droit d'auteur (Bernard Lang, 1998 a). Ces licences des logiciels libres (la Licence Publique Générale GPL GNU, la Berkeley Public Licence...) garantissent que les programmes sont réellement accessibles sans frais pour tous leurs utilisateurs et interdisent les dépôts de licences classiques : "le logiciel libre apparaît fondamentalement comme un système d'anti-brevet, où la propriété n'est pas garantie, mais refusée" (Jean-Michel Dalle, Nicolas Jullien, 2000, p. 6-7). La protection apportée par ces licences se propage par contagion, en garantissant que tout développement supplémentaire sur un logiciel libre sera lui-même un logiciel libre.

Ce que couvrent ces licences, et donc ce qui définit le logiciel libre, c'est uniquement le code source du logiciel, à l'exclusion explicite de tout service lié, y compris même une simple garantie de bon fonctionnement. La production d'un logiciel libre consiste simplement à développer le logiciel, dont le code source est librement accessible. Une personne ou une organisation intéressée par ce logiciel devra effectuer elle-même ou faire réaliser par des

prestataires extérieurs les opérations complémentaires nécessaires (duplication, compilation, édition de liens, installation, paramétrage, adaptation, formation...) pour que le logiciel puisse répondre à ses besoins. Toutefois on peut noter que dans la mesure où les opérations de transformation d'un logiciel-source mis au point en un programme exécutable (code objet) demandent peu d'efforts pour ceux qui disposent des compétences, des logiciels et du matériel nécessaires, il est fréquent de proposer également librement les programmes exécutables correspondants aux configurations informatiques les plus répandues. Il faut également ajouter que les opérations de duplication et de mise à disposition des programmes, dont le coût était déjà très faible par rapport aux coûts de développement du programme, tendent vers zéro avec l'utilisation d'Internet. Enfin de nombreuses sociétés se sont créées, qui proposent tout ou partie des services complémentaires indispensables pour pouvoir utiliser des logiciels libres. C'est du reste l'existence de ces sociétés qui va permettre d'élargir l'utilisation des logiciels libres au-delà de la sphère des utilisateurs experts et équipés, ou qui disposaient en interne des compétences nécessaires.

L'originalité des logiciels libres réside donc dans la fourniture du code source. Dans le cas des autres logiciels (logiciels "privés"⁷⁴), l'entreprise ne vend pas le code source mais le droit d'utiliser un produit partiel dérivé (une version immédiatement exécutable dans un environnement restreint, qui ne peut être revendue, ni modifiée même de façon minimale y compris pour en corriger les défauts : les opérations de décompilation nécessaires pour reconstituer le code source, outre leur difficulté technique, sont généralement interdites par les licences d'utilisation). En d'autres termes, dans le cas des logiciels privés, l'entreprise vend uniquement l'usage d'un produit dont elle conserve le contrôle.

b - ...ce qui est une condition suffisante (mais non nécessaire) à sa gratuité

Il importe tout d'abord de préciser que tous les logiciels gratuits ne sont pas des logiciels libres : certains logiciels sont fournis gratuitement sans que leur code source soit librement accessible. Dans ces situations, la gratuité peut être la conséquence d'une volonté de création désintéressée (*freeware*), ou au contraire être le résultat de stratégies commerciales diverses

⁷⁴ De même que pour les standards (cf. chapitre II) l'expression "logiciel propriétaire" qui est le plus souvent opposé à logiciel libre nous semble peu satisfaisante, un logiciel ne pouvant être propriétaire de quoi que ce soit. Par contre, "logiciel privé" désigne le fait que ce logiciel est la propriété d'un agent économique privé, à la différence d'un logiciel libre qui est "la propriété collective de l'humanité" (Roberto Di Cosmo, Dominique Nora, 1998, p. 144).

d'entreprises à but lucratif. Ces stratégies peuvent avoir pour objectif de faciliter l'imposition d'un produit contre des produits concurrents qui permettra des bénéfices ultérieurs (Microsoft avec Explorer), de rechercher une rentabilité croisée sur des produits complémentaires (Acrobat Viewer d'Adobe⁷⁵), d'obtenir un financement par de la publicité liée au produit (la publicité sur des sites d'accès à Internet auxquels conduisent l'utilisation de certains navigateurs).

Par contre, la libre disposition du code source des logiciels libres implique la gratuité de fait de ces logiciels. Cette caractéristique est souvent contestée, y compris par les promoteurs des logiciels libres. Cette négation de la gratuité des logiciels libres est peut-être liée à l'opinion très répandue dans une société caractérisée principalement par l'extension du règne de la marchandise, que les notions de gratuité et de qualité sont nécessairement antinomiques⁷⁶. Expliquer comme Eric Raymond (1998) que "les logiciels libres ne sont pas l'opposé des logiciels commerciaux" et que "le contraire de libre n'est pas commercial mais propriétaire" nous semble fallacieux : une entreprise propriétaire du code source d'un logiciel peut commercialiser *uniquement* le logiciel ce qui est impossible quand le code source est libre. En effet, le code source d'un logiciel n'est pas simplement le secret de fabrication d'un produit, qui par ailleurs est rarement disponible librement. A partir du code source, il n'existe pas de processus de fabrication à proprement parler, nécessitant des compétences tacites complémentaires et ayant un coût significatif, mais simplement des opérations de compilation, d'édition de liens et de reproduction très facilement et très rapidement assurées par des outils logiciels.

Certes, la progression de la diffusion des logiciels libres s'accompagne du développement de services marchands divers liés à ces logiciels, mais il importe, sur un plan analytique, de différencier ces services des logiciels libres eux-mêmes. Formellement les licences publiques n'interdisent pas de vendre des logiciels libres (à condition de fournir également le code source), et il existe des exemples de sociétés qui, en apparence, vendent des logiciels libres, la plus célèbre étant la Red Hat Software, une des sociétés qui commercialise

⁷⁵ Ce logiciel qui permet de visualiser des documents au format PDF est gratuit, ce qui a permis d'assurer le succès de ce format de fichier puisqu'on est assuré que toute personne pourra facilement en prendre connaissance. Par contre le logiciel de création de documents PDF est payant.

⁷⁶ Par exemple, Gérard Dréan affirme qu'un "logiciel de qualité industrielle ne peut être que payant" (1996 A, p. 246).

Linux. En réalité ces sociétés commercialisent des activités liées au logiciel (sélection de logiciels, réalisation de la copie, distribution, garantie, maintenance, intégration, conseil, installation, assistance technique, développement de solutions spécifiques...), pas le logiciel lui-même⁷⁷ : une personne intéressée par le logiciel seul, peut le télécharger gratuitement ou le compiler à partir de son code source librement accessible. Si certaines de ces sociétés contribuent, à des degrés divers, au développement des logiciels libres sur lesquels sont basés leurs services, il n'est nullement évident que cette situation se pérennise (cf. chapitre VIII).

Les principales caractéristiques des différents produits sont résumées dans le tableau suivant :

Tableau LIII

	Logiciel libre	Logiciel privé payant	Logiciel privé gratuit	Vente de solutions basées sur des logiciels libres
Code source	oui	non	non	oui
Code objet	éventuellement	oui	oui	oui
Services liés	non	oui	éventuellement	oui
Gratuité	oui	non	oui	non

La libre accessibilité du code source est donc une condition suffisante, mais non nécessaire de la gratuité d'un logiciel. Cette discussion sur la caractérisation des logiciels libres comme étant gratuits n'est pas ... gratuite. Elle vise à mettre en évidence le *caractère fondamentalement non marchand de la création des logiciels libres*.

Le modèle libre, que nous caractérisons comme correspondant au "monde de la création" pour le logiciel, est souvent présenté par les promoteurs du logiciel libre comme "reposant sur les mêmes bases que l'économie libérale" (Bernard Lang, 1998 b) et comme se

⁷⁷ L'activité de la Red Hat est du reste résumée ainsi par son PDG Bob Young: "nous testons les versions de Linux disponibles gratuitement, nous ajoutons quelques fonctions de notre cru, nous le mettons dans une boîte en carton avec les mots «Red Hat» écrits dessus, et nous le distribuons pour construire une grande marque connue" (Interview à Libération, 13 octobre 1999). D'autres sociétés vendent des outils et du support technique autour d'un logiciel libre (la société Scriptic fondé par John Ousterhout créateur du langage de script TCL qui est un logiciel libre), des services de conseil autour d'applications (la société suédoise TCX dont le code source de son logiciel système de gestion de base de données, MySQL, est libre) ou des adaptations spécifiques d'un logiciel libre aux besoins des clients (un compilateur C/C++ pour la société Cygnus).

comportant comme "un marché libre" (Eric Raymond, 1998)⁷⁸ ; de même Patrice Aron (1998) parle d'un "modèle d'activité économique plus en accord avec les lois du marché". De façon, à notre avis, contradictoire, les mêmes auteurs soulignent les analogies entre le modèle libre et celui de la recherche scientifique. Ce qui justifie à leurs yeux cette double affirmation, c'est l'existence dans les deux cas d'une concurrence souvent très dure. Il nous semble en revanche que ce qui caractérise "l'économie libérale", dont les principes sont à la base (sous des formes différenciées) des trois autres mondes de production⁷⁹, c'est la production de marchandises dans le but de réaliser un profit ; ce n'est manifestement pas le cas dans le modèle libre où "l'économie monétaire ne joue qu'un rôle réduit" (Bernard Lang, 1998 A). S'il peut effectivement exister des formes de concurrence dans l'économie libérale comme dans le monde de la création, les critères de réussite dans les deux cas sont différents : dans le premier cas, la réussite se mesure en termes de chiffre d'affaires et de profits réalisés, dans le second cas la concurrence prend la forme d'une compétition intellectuelle dont l'enjeu est le prestige qui résulte de la qualité de la création réalisée. Certes l'augmentation de la part de marché d'une entreprise peut résulter de la qualité de ses produits, mais il existe bien d'autres moyens (savoir-faire marketing, puissance de la distribution, vente de produits liés...) pour imposer des produits même de qualité médiocre (cf. supra). Les controverses sur l'évaluation de Microsoft sont caractéristiques de cette opposition : à l'aune des critères marchands, le succès de Microsoft est éclatant, ce que traduit parfaitement ses remarquables résultats financiers, mais cette société est virulemment critiquée par ceux qui se basent sur un critère différent : les qualités technologiques, jugées très insuffisantes, de ses produits (cf. par exemple, Roberto Di Cosmo, 1998).

Il reste à analyser les motivations spécifiques dans une société essentiellement marchande, qui permettent d'expliquer la place particulière qu'ont pu prendre dans l'économie du logiciel, des créations non marchandes, en intégrant le fait que ces productions ne sont pas non plus l'aboutissement d'un projet explicite des pouvoirs publics⁸⁰.

⁷⁸ De façon à notre avis contradictoire, Eric S. Raymond précise dans un article postérieur que "la culture des logiciels libres n'a rien qui ressemble de près ou de loin à de l'argent ou à une économie de marché" (2000, p. 8).

⁷⁹ A l'exception de la partie du monde interpersonnel qui est constituée des services internes aux entreprises.

⁸⁰ Ce qui ne signifie pas que les institutions publiques n'aient pas joué un rôle indirect important comme nous le verrons, notamment dans l'utilisation d'infrastructures matérielles publiques, de temps de travail de

2 - Les motivations pour développer des logiciels libres

Ce qui a incité des informaticiens à développer des logiciels libres, c'est principalement l'inexistence ou la qualité insuffisante des logiciels existants (a), une certaine tradition éthique (b) et la nature particulière de l'activité de développement des logiciels (c).

a - inexistence ou qualité insuffisante des logiciels existants

Dans une situation où les utilisateurs jugent insuffisante la qualité des produits proposés (en raison des mécanismes décrits précédemment), la tentation peut être grande pour des personnes disposant des compétences nécessaires, de développer eux-mêmes les logiciels dont ils ont besoin ; tel est le cas pour certains des acteurs du "monde de la création". Cette tentation peut se transformer en nécessité absolue quand ces personnes ne trouvent pas sur le marché les logiciels qui leur sont nécessaires, par exemple les universitaires américains dans les années 80 pour les logiciels d'accès à Internet (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 92). Cette activité est facilitée par le fait que la création de logiciels réclame peu d'investissements matériels et n'exige aucune infrastructure lourde. De plus, une des particularités de la recherche en informatique est qu'il est fréquent que les chercheurs soient les premiers utilisateurs de leur recherche. Par exemple, Charles Du Granrut montre comment l'équipe de Douglas Engelbart, au Stanford Research Institute, qui durant les années cinquante et soixante élaborait les fenêtres, les images, la souris, le traitement de texte WYSIWYG (What You See Is What You Get), et le système NLS⁸¹, développait d'abord pour son usage personnel de nouveaux outils informatiques qu'elle testait et qui lui permettait par la suite de mettre au point de nouveaux outils (1996, p. 75). Il reste toutefois à expliquer pourquoi, dans certains cas, quand un nouveau logiciel est développé pour répondre à des besoins existants, il l'est sous forme de logiciel libre.

b - une tradition éthique

Une première explication réside dans la proximité entre l'activité de développement de logiciels et la recherche scientifique. Ce phénomène était particulièrement net au début de la constitution de l'informatique comme discipline scientifique, et il s'est maintenu dans certains

personnes payées par l'Etat, et dans la diffusion d'un modèle culturel propice au logiciel libre (de façon quelque peu schématique, le modèle de la recherche scientifique publique).

⁸¹ Le système NLS (On Line System) s'appuyait sur un réseau d'ordinateurs et réalisait déjà certains services comme le courrier électronique ou la téléconférence ; il comprenait un système de références croisées entre différents textes.

domaines où il n'existe pas de séparation nette entre la composante fondamentale et les composantes de recherche appliquée et de développement : on peut citer comme exemples actuels les recherches sur la compression de données ou le cryptage, dont les résultats débouchent de façon quasi immédiate sur des produits opérationnels.

Or, la communauté scientifique est régie par la "convention de la science ouverte" (Partha Dasgupta, Paul David, 1994), qui stipule que les découvertes doivent être rendues publiques dans des conférences et des journaux scientifiques, la réputation récompensant le premier à publier. Dans ce monde, utiliser l'idée d'un autre (en précisant qu'il en est l'auteur) est considéré comme un hommage qui lui est rendu et non comme une exploitation de son travail. Il n'est dès lors pas surprenant pour des produits qui ne sont constitués que de connaissances codifiées, qu'une "tradition de diffusion non marchande de logiciels s'instaure dès le début de l'informatique" (Christian Genthon, Denis Phan, 1999, p. 172), les programmes étant assez librement échangés entre les différents producteurs / utilisateurs sans paiement de redevances⁸². Comme le remarque Gérard Dréan, "l'idée que le logiciel est une création de l'esprit qui doit échapper aux circuits marchands est ancienne et encore bien ancrée dans de nombreux esprits" (1996 A, p. 206). C'est cette idée qui va être reprise dans le modèle du logiciel libre, organisé selon les mêmes principes que la recherche scientifique : circulation libre de l'information qui est critiquée publiquement, contrôle par les pairs, proposition de solutions alternatives, concurrence acharnée entre les équipes (Bernard Lang, 1998 C).

L'événement qui va conduire à la formalisation du modèle du logiciel libre, avec l'invention de la licence GPL par la Free Software Foundation, va être la volonté de créer un Unix libre, alors que le système Unix qui avait été créé et amélioré dans le milieu de la recherche selon des principes de coopération, était en train d'éclater en de multiples variantes commerciales incompatibles (cf. chapitre II).

⁸² Richard Stallman, chercheur au MIT et "inventeur" de la licence GPL, décrit ainsi la situation dans les années soixante dix : "Nous étions dans une communauté de partage du logiciel, qui comprenait plusieurs universités, et parfois des coopérations avec des fabricants d'ordinateurs comme Digital. Si d'aventure, vous aviez vu à l'écran de quelqu'un d'autre quelque chose d'intéressant, vous pouviez lui demander : "Quel programme est-ce ?" et il répondait : "c'est le programme nouveau de quelque chose, le code source est dans le répertoire". Et vous pouviez lancer le programme, l'utiliser. Vous pouviez changer le programme, ajouter d'autres fonctions, couper des morceaux de programmes pour les mettre dans des programmes nouveaux que vous écriviez".

Ce modèle est également soutenu par des motivations idéologiques, qui sont parfois floues⁸³, et dont les différentes contributions rassemblées dans le livre de Olivier Blondeau et Florent Latrive (2000) sont représentatives. Florent Latrive, dans sa présentation estime qu'ils constituent une "coalition improbable" composée de "néolibéraux, libertariens, tiers-mondistes, protomarxistes". On peut y retrouver des références à la théorie de la propriété foncière de Locke appliquée à la "conquête de la Noosphère" ("territoire des idées et espace de toutes les pensées possibles") dans la contribution de Eric S. Raymond (2000) mais également l'évocation d'un "anarcho-communisme réellement existant" (Richard Barbrook, 2000).

Le point commun est la volonté de défendre la "liberté" pour les utilisateurs de logiciels, Richard Stallman (1998) distinguant quatre niveaux de liberté : "la liberté d'utiliser le programme, quel que soit l'usage (liberté 0) ; la liberté d'étudier le fonctionnement du programme et de l'adapter à vos besoins (liberté 1) ; la liberté de redistribuer des copies, et donc d'aider votre voisin (liberté 2) ; la liberté d'améliorer le programme et de diffuser vos améliorations au public, de telle sorte que la communauté tout entière en bénéficie (liberté 3)". C'est cette exigence de conserver l'ensemble de ces libertés qui conduit Richard Stallman à vouloir maintenir l'expression de "*free software*" de préférence à "*open source*", expression suggérée par certains développeurs de logiciels libres qui estimaient "qu'en restant prudent à propos de l'éthique et de la liberté, et en ne parlant que des bénéfiques pratiques et immédiats de certains logiciels libres, ils pouvaient "vendre" le logiciel libre plus efficacement à certains utilisateurs, principalement en entreprise" (1998, p. 2). Or le fait que le code source du logiciel soit ouvert ne suffit pas à garantir que le logiciel restera protégé par une licence "publique" et ne sera pas transformé en un logiciel privé, ce qui supprimerait une partie des libertés de l'utilisateur (cf. chapitre VIII).

Il est patent que "certains projets de logiciels libres doivent leurs origines à des sentiments anti-proprétaires et de lutte contre des monopoles comme ceux de Microsoft et [que] cela a certainement donné une impulsion significative (voire décisive) au

⁸³ Christian Genthon et Denis Phan parlent d'un "discours libertarien à la fois libertaire et libéral" (1999, p. 181).

développement du logiciel libre" (Jean-Michel Dalle, Nicolas Jullien, 2000, p. 7)⁸⁴. Nous avons analysé (cf. chapitre III) en quoi les logiciels possédaient "naturellement" les déterminants des biens collectifs et les efforts (techniques et juridiques) pour les transformer en un produit commercialisable. A l'inverse l'approche des partisans des logiciels libres consiste à revendiquer et à défendre le caractère de bien collectif des logiciels, au nom de l'exigence du partage de la connaissance. L'analyse économique postule que sont généralement adoptés des comportements de "*cavalier libre*" par rapport aux biens collectifs. Nous verrons qu'en raison de la nature de cette activité, des comportements de contribution "désintéressée" existent, confirmant ce que Albert O. Hirschman disait des biens collectifs résultant d'une action de groupe : "comme l'issue et l'objectif de l'action collective sont d'ordinaire un bien public offert à tous, la seule manière pour un individu d'accroître le bénéfice que lui procure l'action collective est d'augmenter *son apport personnel (input)*, l'effort qu'il consent pour la politique publique à laquelle il se rallie. Bien loin de se dérober et d'essayer d'obtenir un billet gratuit, l'individu qui cherche réellement à maximiser son utilité s'efforcera d'être aussi activiste que possible" (1986, p. 101).

Dans l'économie du logiciel qui est une économie d'abondance⁸⁵, c'est la "culture du don" qui s'impose : "le statut social n'est pas déterminé par ce que vous contrôlez, mais par ce que vous *donnez* (...). L'abondance crée une situation où la seule évaluation possible de la réussite dans cette compétition est la réputation que chacun acquiert auprès de ses pairs" (Eric S. Raymond, 2000, p. 10). "Les participants rivalisent pour le prestige en donnant du temps, de l'énergie, et de la créativité" (idem, p. 23).

c - la nature particulière de l'activité de développement des logiciels

Par bien des aspects (infinie diversité, caractère immatériel, facilité de reproduction) la programmation s'apparente à l'écriture (cf. chapitre I), même si cette forme d'écriture est plus proche d'une démonstration mathématique que de la création d'une œuvre littéraire.

⁸⁴ Par contre nous ne partageons pas l'avis de ces auteurs selon lesquels "ce type de motivations "affectives" a largement cessé d'être dominant dans les incitations à participer au développement du logiciel libre" (idem).

⁸⁵ "Que signifie le terme "propriété" lorsque ce qui est possédé est duplicable à l'infini, hautement malléable, et que la culture environnante n'est plus capable de faire appliquer des lois, et n'est plus dans une situation économique où les ressources sont limitées" s'interroge Eric S. Raymond (2000, p. 5).

Selon Pierre Lévy (1992), la programmation relève même des beaux-arts. Le plaisir de créer et d'être reconnu peut constituer une motivation suffisante à la production de logiciels. Eric S. Raymond parle de la "satisfaction purement artistique de mettre au point et de faire fonctionner un bon logiciel" (2000, p. 10)⁸⁶. Bernard Lang (1998 A) va jusqu'à effectuer une analogie avec les motivations de la création musicale. De même, Marvin Minsky, grand spécialiste de l'intelligence artificielle, estime qu'il n'y a pas une grande différence entre les *hackers*⁸⁷ et ceux qui se consacrent entièrement à un travail créatif, comme les poètes, les artistes ou les écrivains (Philippe Breton, 1990, p. 102). Dans les deux cas, on retrouve les caractéristiques de l'état de grandeur du monde de l'inspiration (Luc Boltanski et Laurent Thévenot, 1991).

Frederick P. Brooks estime que la "programmation est une joie parce qu'elle satisfait nos instincts créatifs les plus profonds et qu'elle permet à la sensibilité humaine de s'exprimer" (1996, p. 7). Il détaille les plaisirs qu'apportent la programmation : "pur plaisir de la création (...), plaisir de créer des choses utiles à autrui (...), fascination qu'exercent les mécanismes compliqués faits de multiples pièces mobiles, et qu'on aime regarder évoluer en cycles subtils, déroulant inexorablement les conséquences de leurs principes de construction (...), bonheur de toujours apprendre, qui découle de la nature non répétée de la tâche (...), joie de travailler sur un support aussi souple [où] le programmeur, comme le poète, manie des abstractions voisines de la pensée pure" (idem, p. 6). Selon Eric Raymond (1998), la motivation principale des développeurs des logiciels libres est ce qu'il appelle "*Ego Gratification*" (la gratification de l'ego) : "la fonction d'utilité (...) qu'ils maximisent n'est pas classiquement économique, c'est l'intangible de leur propre satisfaction personnelle et leur réputation"⁸⁸.

⁸⁶ Ceci peut également expliquer la production de logiciels libres par des informaticiens à la "créativité refoulée" dans les entreprises commerciales où ils travaillent : "trop souvent on voit des développeurs de logiciels se morfondre à produire des programmes dont ils n'ont pas besoin et qu'ils n'aiment pas" (Eric Raymond, 1998).

⁸⁷ Les *hackers* (de *to hack* piocher, labourer, le *hack* en informatique ayant pris le sens d'une invention astucieuse, une solution élégante à un problème) sont de vrais "mordus" de l'informatique, au sens où ils y consacrent l'essentiel de leur énergie vitale (Philippe Breton, 1990, p. 97). Les principales caractéristiques des hackers sont l'habileté technique et le plaisir de résoudre des problèmes et de dépasser les limites.

⁸⁸ Un exemple intéressant dans un domaine voisin (la création de jeux vidéo) de motivations du même ordre, mais habilement exploitées commercialement, est le concours organisé chaque année par Nintendo au Japon. Le gagnant de ce concours, très prisé, ne reçoit rien si ce n'est la gloire d'une diffusion massive de sa création, qui est distribuée sur cartouche (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 121).

Pour Philippe Breton, "l'informatique exalte l'acte créateur, et plus particulièrement cette forme de création qui ne fait pas appel au corps mais à l'esprit" (1990, p. 52) : "la culture de l'informatique est un appel vibrant à la création humaine, notamment la création technique, en lieu et place d'une création divine absente" (idem, p. 48). Analysant la "tribu informatique", il montre que la hiérarchie au sein de celle-ci est moins fondée sur le statut que sur la compétence.

Or la participation au développement de logiciels libres peut permettre d'exhiber celle-ci⁸⁹. L'identification des auteurs de logiciels libres est scrupuleusement garantie par les licences publiques : les lignes de codes sont "signées" par leurs auteurs, le nom du développeur étant inscrit près des parties de code source sur lesquelles il a travaillé, et dans la plupart des logiciels libres il existe un fichier "credits" qui recense les principaux contributeurs du logiciel et leurs apports. De plus, les licences publiques intègrent des dispositifs visant à préserver, en cas de modifications du logiciel, "la réputation de l'auteur original".

Cette reconnaissance de la contribution de chacun des créateurs des logiciels libres a des conséquences sur la qualité des logiciels produits : "la disponibilité du code source met aussi en jeu l'orgueil du programmeur, qui sait qu'il va être jugé par ses pairs. Et il existe pour un informaticien peu de satisfactions personnelles aussi grandes que celle d'avoir contribué à écrire un programme qui est apprécié, utilisé, repris et amélioré pendant dix ans par des milliers de programmeurs et des millions d'utilisateurs, le tout pour ses mérites propres" (Roberto Di Cosmo et Dominique Nora, 1998, p. 164). En fin de compte, "les logiciels libres sont faits par des artisans amoureux de leur art. Ce ne sont pas des produits fabriqués en grande série qui doivent respecter un calendrier, quitte à comporter des défauts" (Linus Torvalds dans Le Monde du 27-28 septembre 1998)⁹⁰.

On peut ajouter que participer au développement de logiciels libres peut contribuer à augmenter ses compétences, et que pour les principaux animateurs des projets qui connaissent

⁸⁹ Analysant les motivations des *hackers*, Nicolas Auray montre que, contrairement aux idées reçues, ceux-ci loin de se limiter à un "tête à tête solitaire et autistique avec un objet" (p. 191) recherchent la valorisation publique.

⁹⁰ En sens inverse, Toran Demarco et Timothy Lister estiment que développer des produits de piètre qualité (car contraints par les délais et les exigences immédiates des clients) peut avoir des effets négatifs sur l'activité des programmeurs, car ils sont peu satisfaits de leur comportement et de leur efficacité (1991, p. 30-31).

un certain succès, des effets de notoriété peuvent permettre une valorisation pécuniaire ultérieure de leurs compétences. Eric S. Raymond indique ainsi que "parfois la réputation acquise (...) peut se répandre dans le monde réel et avoir des répercussions financières significatives [par] l'accès à une offre d'emploi plus intéressante, à un contrat de consultant, ou aiguïser l'intérêt d'un éditeur", mais il précise que "ce type d'effet de bord est rare et marginal (...) ce qui est insuffisant pour en faire une explication convaincante" (2000, p. 8-9).

3 - des qualités spécifiques en raison du mode de développement adopté

A priori les logiciels créés dans le monde de la création apparaissent dans une situation peu favorable par rapport aux dimensions critiques analysées : la productivité dans la production de logiciels dépendant principalement de l'importance de la diffusion par la répartition sur l'ensemble des exemplaires produits des coûts de conception, elle reste tributaire du succès de ces logiciels qui n'est nullement garanti⁹¹, la fiabilité des premiers produits créés est fréquemment problématique et leur adéquation aux besoins des utilisateurs n'est pas toujours avérée. Mais pour les logiciels libres qui connaissent un certain succès initial, le fait que leur code source soit public peut permettre une amélioration rapide de la fiabilité des logiciels, des modifications permanentes des produits initiaux pouvant aboutir à des utilisations diversifiées et parfois très éloignées des objectifs du créateur initial, et donc une diffusion importante. Le processus particulier de développement des logiciels libres leur confère des caractéristiques techniques de haut niveau, qui expliquent le succès de certains d'entre eux (Roberto Di Cosmo et Dominique Nora, 1998, Bernard Lang, 1997, Patrice Aron, 1998) : flexibilité (modification du logiciel en fonction de l'évolution des besoins), portabilité (possibilité d'adapter le logiciel à la configuration matérielle utilisée qui garantit une indépendance par rapport aux producteurs de matériel), pérennité sur le long terme⁹², compatibilité (dans la mesure où ces logiciels sont construits sur des standards ouverts).

⁹¹ Cependant la faiblesse de la productivité doit être rapportée à l'originalité du produit créé et des méthodes utilisées, qui peut avoir des conséquences très positives sur l'ensemble des producteurs de logiciels. Il faut également ajouter que la production de logiciels libres n'a pas à supporter les dépenses de marketing dont l'importance est croissante dans la production des logiciels commerciaux.

⁹² A la différence des logiciels commerciaux où la pérennité est souvent problématique, un éditeur "faible" risquant de disparaître et un éditeur "fort" pouvant imposer la diffusion de nouvelles versions et l'abandon des anciennes en fonction de ses intérêts, la pérennité du logiciel libre est garantie par la possibilité de continuer à développer un logiciel qui n'intéresse plus ses créateurs initiaux (l'exemple de "pop client" décrit par Eric Raymond (1998) en est une bonne illustration).

a - Un mode de développement original

Le mode de développement des logiciels libres pousse à l'extrême certaines tendances observées dans les sociétés qui se sont imposées sur le marché des progiciels (cf. supra), en étant fondé sur un "ensemble de coutumes de coopération opposé à la direction par coercition" (Eric S. Raymond, 1998). Selon Jacques Printz, "les stratégies de comportement où les personnes coopèrent librement, stratégies dites donnant-donnant (...), sont les plus efficaces sur le moyen terme. Dans ce type de comportement, il y a reconnaissance permanente et valorisation de la contribution d'un membre de l'équipe par les autres" (1998, p.82). Ce mode de développement constitue également une illustration d'un modèle plus général en émergence, appelé "connaissances distribuées", où sont mis en valeur des liens locaux, des relations aux équipements, des coopérations, des dynamiques émergentes, en opposition aux modèles centrés sur les hiérarchies, les plans et les capacités individuelles (Laurent Thévenot, 1997, p. 225).

Nous avons vu que le développement de beaucoup de logiciels libres avait été initié par des informaticiens pour répondre à des besoins non ou mal satisfaits par les progiciels commerciaux existants. Le choix de développer ses logiciels sous forme de logiciels libres correspond également à une recherche d'efficacité pour résoudre le problème qui se pose à l'utilisateur. En effet, ce problème a de fortes chances d'être commun à un certain nombre d'utilisateurs potentiels compétents⁹³ entre lesquels l'information circule rapidement (communauté structurée par des réseaux⁹⁴) et il est donc possible de transformer ces utilisateurs en co-développeurs permettant "une amélioration rapide du code et un débogage efficace" (Eric Raymond, 1998). L'élimination des bogues, qui constitue fréquemment la majeure partie de l'activité de conception d'un logiciel, peut être particulièrement performante en parallélisant le débogage : "étant donné un ensemble de bêta-testeurs et de co-développeurs suffisamment grand, chaque problème sera rapidement isolé, et sa solution semblera évidente à quelqu'un" car chacun a "une perception du problème, des outils d'analyse, un angle

⁹³ Les meilleures réalisations "commencent en tant que solutions personnelles aux problèmes de tous les jours rencontrés par leur auteur, et elles se répandent parce que ce problème est commun à de nombreuses personnes. (...) Pour résoudre un problème intéressant, commencez par trouver un problème qui vous intéresse" (Eric Raymond, 1998).

⁹⁴ Les réseaux humains existants ont été considérablement renforcés et étendus par l'extension et l'interconnexion des réseaux informatiques (Internet), qui ont grandement facilité les échanges pour développer des logiciels libres (rapidité et baisse des coûts). Réciproquement les logiciels libres ont joué un rôle décisif dans le développement d'Internet.

d'attaque qui lui sont propres" (Eric Raymond, 1998) à la différence des logiciels commerciaux qui, le plus souvent, "sont revus par des personnes très proches des auteurs, et qui commettent les mêmes erreurs" (Linus Torvalds dans *Le Monde* du 27-28 septembre 1998). Pour les membres de ce que Philippe Breton nomme "la tribu informatique", "les erreurs paradoxalement sont très recherchées car elles sont l'occasion d'une interaction intense, qui mobilise toutes les facultés mentales et procurent une étonnante sensation de "corps à corps" –si l'on peut dire – avec la machine" (...)."Les constants perfectionnements des programmes sous la forme de nombreuses versions successives, fournissent l'occasion d'un renouvellement permanent de la communication entre créateurs et utilisateurs, via l'ordinateur" (1990, p. 70) et peut permettre de retrouver l'enthousiasme devant un nouveau programme imprévisible, qui après connaissance engendre la lassitude.

Ce mode de travail peut s'étendre au-delà des tests, jusqu'au développement en parallèle de différentes propositions indépendantes dont la meilleure est retenue⁹⁵. Cette méthode que Christian Genthon et Denis Phan qualifient de "développement distribué" (1999, p. 179) est une méthode décentralisée, fondée sur la sélection compétitive de solutions locales mises en concurrence, qui peut être plus efficace qu'une solution centralisée⁹⁶. Elle peut être complétée par le fait que "la majeure partie des phases d'architecture, d'implémentation et de réalisation d'un logiciel peuvent être menées en parallèles" (Frederick P. Brooks, 1996, p. 203), à la condition que ce logiciel soit construit selon une architecture ouverte, documentée et modulaire, ce qui est considéré unanimement comme un facteur décisif de qualité (Bernard Lang, 1997, p. 5).

Ce modèle de développement du logiciel, que Eric Raymond (1998) appelle le modèle du bazar⁹⁷, a fait ses preuves notamment pour le développement de grands projets complexes

⁹⁵ Cette méthode n'est pas sans rappeler la "logique organisationnelle de la redondance" dont Ikujiro Nonaka et Hirotaka Takeuchi ont montré l'efficacité (1997, p. 32).

⁹⁶ Christian Genthon et Denis Phan établissent une analogie avec des recherches en intelligence artificielle, qui montrent que des méthodes décentralisées de résolution de problèmes par systèmes "multi-agents" peuvent se révéler d'autant plus efficaces relativement aux méthodes centralisées que la complexité augmente (1999, p. 179).

⁹⁷ Ce modèle - opposé "au modèle de la cathédrale" - ne signifie pas qu'il n'existe pas une forme d'organisation dans le développement d'un logiciel libre. Au minimum il est nécessaire qu'existe une instance de décision (leader du projet reconnu par la communauté appelé "dictateur bienveillant" par Eric Raymond, procédure de vote comme dans le cas d'Apache, rotation du coordinateur au sein d'un cercle de co-développeurs aguerris dans le cas de Perl) pour trancher entre les différentes propositions.

comme l'écriture du système d'exploitation Linux. Combiné au développement d'Internet, il permet de mettre à contribution l'attention et la puissance de réflexion de communautés entières de développeurs dispersés géographiquement, "qui peuvent mettre sur le problème un temps humain cumulé plus important de plusieurs ordres de grandeurs" (Eric Raymond, 1998) que n'importe quelle entreprise commerciale. Les nouvelles pratiques de plus en plus utilisées par tous les producteurs de logiciel (prototypage rapide, programmation évolutive, mobilisation des compétences des utilisateurs) rendent ce modèle de développement libre particulièrement efficace.

La force du modèle du libre tient en grande partie à la qualité des relations entre producteurs et utilisateurs du logiciel, pouvant aller jusqu'à une réversibilité possible des rôles⁹⁸, ce qui n'est pas sans rappeler les relations entre producteurs et consommateurs d'informations sur Internet. Marie de Besses souligne que "le système du libre est (...) en définitive un système de circulation de connaissances, encore plus que de produits" (1999, p; 276) à l'opposé "des relations de circulation du produit, dépourvues volontairement de circulation d'informations" des logiciels privés (1999, p. 276-277). Ce système génère un "espace d'apprentissage et de production interactive d'un grand dynamisme", l'existence de "relations directes et interactives entre concepteurs et utilisateurs (...) donnant toujours lieu (...) à acquisition de connaissances nouvelles" (idem). "Le concepteur initial tire parti des critiques, et des éventuelles améliorations de ses utilisateurs" (idem, p. 274). L'implication des utilisateurs dans l'amélioration du produit initial s'explique également parce qu'ils "tirent parti de l'accès à l'ensemble des éléments du produit technique, et donc de la compréhension du travail du concepteur" (idem).

Certes il importe de distinguer parmi les utilisateurs des logiciels libres, les utilisateurs actifs qui participent à son développement, des "utilisateurs passifs" qui se contentent d'utiliser des logiciels libres, de signaler les défauts et d'attendre les mises à jour. "La viabilité des logiciels libres dépend principalement de l'existence d'une forte communauté d'utilisateurs actifs" (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 102). Ce qui motive des utilisateurs

⁹⁸ "Traiter vos utilisateurs en tant que co-développeurs est le chemin le moins semé d'embûches vers une amélioration rapide du code et un débogage efficace" explique Eric S. Raymond (1998).

passifs à devenir des utilisateurs actifs⁹⁹, c'est outre "la perspective autogratifiante de prendre part à l'action et d'être récompensés par la vue constante (et même quotidienne) des améliorations de leur travail" (Eric Raymond, 1998), le fait qu'en raison de la nature des logiciels, les améliorations suggérées peuvent profiter directement à l'ensemble des utilisateurs, un logiciel modifié pouvant être directement utilisable sans investissement supplémentaire¹⁰⁰.

b - Un monde innovatif

Le rôle des acteurs du "monde de la création" dans le développement des innovations n'est pas spécifique au logiciel et se retrouve dans l'ensemble de l'informatique depuis les premiers calculateurs expérimentaux construits par des universités ou des centres de recherche pendant et après la deuxième guerre mondiale, jusqu'à l'émergence des machines massivement parallèles, en passant par les découvertes techniques qui permirent la naissance de la microinformatique. Les deux domaines d'innovations (matériel et logiciel) sont par ailleurs liés, comme l'illustrent particulièrement les innovations développées au Palo Alto Research Center de Xerox : les icônes et les fenêtres en 1972, le langage objet Smalltalk en 1972, le premier ordinateur individuel Dynabook ou Alto en 1973, le réseau local Ethernet en 1973. Mais dans le cas du logiciel (à la différence du reste de l'informatique) les inventions peuvent déboucher directement sur des produits diffusés massivement sans la nécessité de passer par un processus de production spécifique distinct du processus de création.

La plupart des innovations radicales concernant les logiciels, tant en termes de produits que de méthodes de production, ont été effectuées dans le "monde de la création". En effet, dans la production marchande de logiciels, les véritables innovations¹⁰¹ sont plus rares, et leur application peut être lente pour plusieurs raisons. Tout d'abord, il est difficile de prévoir les évolutions dans l'ensemble de l'informatique et il existe un risque important d'introduire une

⁹⁹ Cette possibilité n'existe pas pour un logiciel privé. Richard Stallman raconte l'anecdote suivante qui contribua à l'inciter à créer des logiciels libres : face à une imprimante qui se bloquait, il était dans l'impossibilité de modifier le logiciel qui pilotait l'imprimante pour résoudre le problème.

¹⁰⁰ D'où le conseil de Eric Raymond (1998) : "distribuez vite et mettez à jour souvent".

¹⁰¹ Il est nécessaire de distinguer les véritables innovations de la mise sur le marché des nouvelles versions d'un produit existant. Si celles-ci s'effectuent au contraire à un rythme accéléré, leur succès repose plus sur des stratégies commerciales (apparence, introduction de nouvelles fonctionnalités très peu utilisées, modification des formats de fichiers...) que sur de réelles améliorations.

innovation qui sera vite dépassée par une autre. Ensuite, il faut tenir compte du fait que les imitateurs peuvent tirer plus de profit que les initiateurs d'une innovation¹⁰² et qu'une entreprise en position de force peut avoir intérêt à retarder l'introduction d'innovations¹⁰³. Enfin, l'importance des effets d'apprentissage peut expliquer pourquoi des méthodes de résolution de problèmes à long terme (productivité, délais) ne sont pas adoptées, car elles aggravent ces mêmes problèmes à court terme, ce qui peut être rédhibitoire pour des entreprises soumises à des contraintes de rentabilité à court terme (par exemple l'adoption de nouvelles méthodes, de nouveaux langages).

Une explication de la fréquence des innovations dans les logiciels libres vient de la publication de leur code source. "Une des lois fondamentales de l'économie de la créativité est, en effet, que plus cette créativité est accessible, plus elle engendre de créativité. Les découvertes, les innovations, bref les idées sont incrémentales, donc favorisées par l'accès aux idées existantes" (Jean-Michel Dalle, Nicolas Jullien, 2000, p. 5). Bernard Lang (1998 A) va jusqu'à affirmer que "le développement des logiciels est de même nature que celui des théories mathématiques. Or la science (...) s'accommode mal du secret et des barrières qui sont le pain quotidien des développements industriels. Les bonnes spécifications (...) et les bonnes réalisations (...) ne s'élaborent que lentement par un développement social ouvert d'évaluation, de confrontation et de collaboration". C'est du reste la volonté de compenser par des innovations technologiques, l'affaiblissement commercial de son logiciel de navigation, Communicator, face à Internet Explorer de Microsoft, qui a conduit Netscape à "ouvrir" le code source de ce logiciel : "en libérant la puissance créatrice des milliers de programmeurs qui opèrent sur Internet et en intégrant leurs meilleurs enrichissements aux futures versions du Communicator, nous pensons pouvoir amener les innovations technologiques sur le marché des navigateurs à un niveau d'excellence jamais atteint" a déclaré Jim Barksdale, PDG de Netscape.

¹⁰² Notamment quand l'imitateur dispose d'avantages compétitifs sur des actifs complémentaires. On peut considérer que le marketing fait à l'heure actuelle partie de ces actifs complémentaires décisifs sur les marchés des logiciels.

¹⁰³ "IBM a traditionnellement disposé d'un portefeuille impressionnant de technologies de pointe et de produits innovants, mais ne les a généralement introduits sur le marché que sous la pression de la concurrence, donnant ainsi l'image d'une firme retardataire sur le plan technique" (Gérard Dréan, 1996 A, p.298). Par exemple, IBM a été le premier inventeur des processeurs RISC mais ne s'en est pas servi commercialement (Jacques Printz, 1998, p. 339).

Enfin, la qualité des interactions entre les développeurs et les utilisateurs (cf. infra) est un facteur favorable à l'innovation selon le modèle mis en évidence par N. Rosenberg (1983). De plus, "comme les utilisateurs sont aussi les innovateurs – et comme les innovateurs sont aussi les utilisateurs – ces programmes sont développés pour résoudre des problèmes qui se posent réellement aux utilisateurs" (Jean-Michel Dalle, Nicolas Jullien, 2000, p. 4).

c - L'articulation avec les standards

Nous avons souligné l'importance décisive des questions de standardisation en informatique et les potentialités du monde de la création pour promouvoir une standardisation dynamique, par opposition aux standards "propriétaires" imposés par des entreprises privées et aux difficultés à émerger des normes publiques (cf. chapitre II et François Horn, 1999 A). Or, le plus souvent les standards développés par des organisations fondées sur le modèle libre s'appuient sur des logiciels libres ; par exemple, le W3C (World Wide Web Consortium) propose autour de chacune des normes qu'il souhaite promouvoir un jeu de logiciels libres capables de les exploiter (Jean-Paul Smets-Solanes, 1998). En même temps, l'existence de standards publics et documentés facilite le développement de logiciels libres. Ces standards (protocoles et formats de données) sont naturellement scrupuleusement respectés par les développeurs de logiciels libres, à l'opposé des pratiques de certains producteurs de logiciels commerciaux (cf. infra).

d - Succès et limites des logiciels libres

Le mode de développement des logiciels libres permet de comprendre pourquoi certains d'entre eux ont atteint un très haut niveau de *fiabilité*. C'est la conclusion à laquelle est parvenue une étude statistique américaine qui indique que les logiciels libres sont souvent les plus fiables (citée par Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 105). C'est notamment dans le domaine de la sécurité que les logiciels libres se révèlent supérieurs : "Les logiciels libres n'ont pas la complexité des logiciels commerciaux, et l'ouverture de leur code font que la découverte et la correction d'erreurs parfois graves en matière de sécurité sont possibles" affirme Hervé Schauer consultant en sécurité informatique (aux troisièmes rencontres de l'ISOC)¹⁰⁴.

¹⁰⁴ Un exemple récent est le cas du plus célèbre des logiciels de cryptographie, *Pretty Good Privacy* développé à partir des travaux de Philip Zimmermann. Selon certains experts, la commercialisation de nombreuses versions de ce logiciel par la société NAI avec ajout de nouvelles fonctionnalités non

Les logiciels libres se sont révélés bien adaptés à certains besoins de la communauté informatique. Les difficultés pour appréhender les véritables besoins des utilisateurs et la tendance des informaticiens à privilégier les caractéristiques techniques de leur réalisation sur ses caractéristiques d'usage – deux facteurs majeurs pour expliquer le manque d'adéquation aux besoins des utilisateurs (cf. chapitre V) – ne jouent pas quand les premiers utilisateurs sont les développeurs des logiciels. De plus la connaissance du code-source du logiciel peut permettre une meilleure maîtrise de l'utilisation du logiciel grâce à la compréhension des mécanismes sous-jacents. Il est évident que par contre la connaissance du code-source n'est d'aucune utilité directe¹⁰⁵ pour un utilisateur non informaticien. De façon plus générale, les logiciels libres ont plus de difficultés pour répondre aux besoins de simples usagers, ne détenant pas de compétences techniques informatiques, une des explications étant qu'écrire des lignes de code permettant de rendre un logiciel plus facile à utiliser ou à installer n'est pas ce qui est le plus valorisé dans la communauté informatique.

Par contre, quand ce logiciel va être essentiellement destiné à des informaticiens (par exemple un système d'exploitation pour un serveur), les améliorations qualitatives permanentes des logiciels libres permettent d'expliquer comment un logiciel libre peut réussir à s'imposer sur un marché dominé par un logiciel privé en surmontant le verrouillage technologique dû aux rendements croissants d'adoption¹⁰⁶. C'est ce que démontrent Jean-Michel Dalle et Nicolas Jullien (2000) en utilisant un modèle de concurrence technologique (l'exemple analysé est Linux contre Windows 2000¹⁰⁷) intégrant des externalités globales et locales. Les externalités globales sont plus fortes dans le cas du logiciel libre non pas pour des raisons d'ordre technologique (de nombreux travaux analysés dans le chapitre II ont montré qu'une technologie qui s'est imposée sur un marché peut maintenir sa domination même en présence de solutions technologiquement supérieures), mais comme conséquences du modèle

indispensables, s'est accompagné de l'affaiblissement de la sécurité cryptographique du programme, à l'inverse de la version libre de *Pretty Good Privacy*, GnuPGP (Transfert.net, 21 septembre 2000).

¹⁰⁵ Directement, car indirectement la disponibilité du code-source donne la possibilité à d'autres personnes (ou sociétés) d'analyser ce logiciel, et apporte donc une certaine indépendance par rapport aux développeurs du logiciel.

¹⁰⁶ On peut considérer que la diffusion d'un logiciel libre s'apparente à une trajectoire technologique dans un "environnement de sélection non marchand" (R. Nelson, S. Winter, 1975) où ce n'est pas la perspective du profit qui oriente le choix technologique.

¹⁰⁷ Rappelons que malgré les efforts consentis pour promouvoir OS/2, IBM avait échoué dans sa tentative de concurrencer Windows.

économique du logiciel libre : celui-ci fait que les externalités d'adoption sont "redistribuées aux utilisateurs" sous forme d'améliorations du logiciel car "le cœur du modèle libre repose sur la coopération dans la communauté" (p. 12), au lieu d'être "appropriées en profit" (rente de monopole) dans le cas d'un logiciel privé¹⁰⁸. Pour que ces externalités globales supérieures dans le cas des logiciels libres qui font qu'ils "s'améliorent plus efficacement et plus rapidement" (p. 12), débouchent sur le succès de ceux-ci, il est nécessaire qu'existent en plus des externalités locales supérieures permettant de déclencher une nouvelle dynamique de diffusion, que les auteurs identifient comme étant la conséquence "du prosélytisme des adopteurs de Linux" (p. 13).

En ce qui concerne la *productivité* pour la production des logiciels libres, elle dépend principalement de l'importance de leur diffusion par la répartition sur l'ensemble des exemplaires produits, des coûts de conception de l'original. Ces coûts sont difficiles à estimer dans la mesure où ils correspondent à des participations extrêmement variées : chercheurs et enseignants pour qui la création de logiciels peut faire partie de leurs activités de recherche ou d'enseignement, travail gratuit fourni par des étudiants ou des salariés en dehors de leur temps de travail, informaticiens qui profitent directement du logiciel développé¹⁰⁹ et enfin développeurs financés par certaines sociétés qui commercialisent des solutions basées sur des logiciels libres. Quant à la diffusion du logiciel, il importe de noter qu'il peut se prêter à des utilisations totales ou partielles complètement imprévues à l'origine¹¹⁰. Cette application possible à des domaines différents vient du caractère universel et donc diffusant des technologies du traitement de l'information (Michel Delapierre, Jean-Benoît Zimmermann, 1991, p.47). Ici également, les facilités de modification et de réutilisation que permettent la libre disponibilité du code source des logiciels libres facilitent ce processus¹¹¹.

¹⁰⁸ "Quand les externalités d'adoption ne sont pas redistribuées, même une technologie dominante peut être mise en danger par un nouvel entrant qui, au contraire, les redistribue généreusement" (p. 13)

¹⁰⁹ Le cas le plus connu est celui d'Apache, logiciel de serveur Web, développé par des gestionnaires de site Web.

¹¹⁰ "Tout outil doit être utile par rapport aux utilisations qu'il a été prévu d'en faire. Mais on reconnaît un outil vraiment excellent au fait qu'il se prête à des usages totalement insoupçonnés" (Eric Raymond, 1998). Un exemple est la conception d'outils de traitement d'images, destinés au départ à manipuler des photos issues de données satellites dans les laboratoires scientifiques, qui ont donné naissance à des logiciels de retouche d'images grand public.

¹¹¹ On peut établir une analogie avec l'analyse de l'innovation technologique effectuée par Michel Callon comme un "processus tourbillonnaire" qui peut durer longtemps : "On teste un premier prototype, puis on remet le

Les logiciels libres se sont développés quasi spontanément dans certains domaines : des logiciels qui au départ sont principalement autoconsommés par les producteurs eux-mêmes et qui nécessitent pour leur mise au point des compétences spécifiques (dont l'exposition sera valorisée par les pairs) comme les logiciels liés au développement d'Internet, préférablement à des produits plus banalisés (progiciels bureautiques, ou de gestion, applications "métiers"...). Leur extension à l'ensemble des logiciels et des utilisateurs constitue un défi qui ne sera pas forcément relevé spontanément (cf. chapitre VIII).

D - LE MONDE DE LA PRODUCTION FLEXIBLE DES LOGICIELS

Le monde de la production flexible représente d'une certaine façon une tentative pour trouver un compromis entre l'attrait d'une production sur mesure caractéristique du monde interpersonnel, et l'efficacité en termes de productivité du monde fordiste. Plus difficile à délimiter, notamment dans la mesure où il entretient des rapports qui sont plus de complémentarité que de concurrence avec les autres mondes, il recouvre dans l'économie du logiciel deux réalités : d'une part la fourniture avec des logiciels standards de services sur mesure, activité qui n'est pas nouvelle mais qui a pris de nouvelles formes avec le développement des logiciels libres (1) ; d'autre part, la production de logiciels sur mesure à partir de composants standardisés, qui correspond à une évolution plus récente, et encore embryonnaire mais qui semble riche de potentialités (2).

1 - Des logiciels standards accompagnés de services sur mesure

a - Des prestations réalisées à partir de progiciels commerciaux...

Des entreprises peuvent fournir autour d'un progiciel des services variés de formation, d'aide à l'installation et à l'utilisation, et de maintenance. Ces services, qui doivent être adaptés aux compétences et aux besoins spécifiques des utilisateurs, sont nécessaires comme le montre le fait que "les utilisateurs ne se servent en moyenne que de 20 % des fonctionnalités d'un progiciel" (Eurostaf, 1996, p.79) et leur existence peut constituer un argument décisif dans les affrontements concurrentiels entre plusieurs progiciels. Ces prestations peuvent être

produit en chantier ; on teste un deuxième prototype, un troisième, un quatrième jusqu'à ce que l'on obtienne un produit satisfaisant, c'est à dire capable de capter une demande qu'il a contribué à définir et à stabiliser. Chaque boucle peut conduire à des révisions radicales, par exemple à relancer des recherches scientifiques fondamentales ou à considérer des marchés différents de ceux initialement visés. L'irréversibilisation est lente car le processus d'adaptation est progressif et itératif et les investissements réalisés peu coûteux" (Michel Callon, 1994 A, p 14).

fournies par des sociétés indépendantes mais souvent liées au producteur du progiciel, voire par l'éditeur du progiciel lui-même pour qui elles peuvent constituer une activité complémentaire¹¹² et une réponse partielle au problème des "piratages" des progiciels¹¹³.

Pour certains progiciels, comme par exemple les progiciels de gestion d'entreprises, le paramétrage du progiciel et son intégration au système d'information constituent un travail important¹¹⁴. Cette activité est réalisée par des prestataires spécialisés, comme les sociétés de services qui se sont développées autour des progiciels de gestion de SAP, pour des PME qui ne possèdent pas de services informatiques internes ou pour des entreprises qui ont fait le choix d'externaliser tout ou partie de leur informatique (cf. chapitre VI).

Enfin se développent rapidement des prestations *d'intégration de systèmes* qui consistent à fournir un système complet intégrant matériels, logiciels et services. De telles prestations permettent de combler le "vide informationnel entre une offre très éclatée et une demande très marquée par l'incertitude des choix organisationnels et technologiques" (Claire Charbit, Jean-Benoît Zimmermann, 1997, p. 17). Elles constituent un secteur très dynamique (Eurostaf, 1996, p.69-70) sur lequel se positionnent de nombreux acteurs : de grandes S.S.I.I. comprenant des structures spécialisées dans les différents domaines d'applications et le plus souvent insérées dans des réseaux constitués de liaisons financières ou de partenariat avec des sociétés de conseil, des producteurs de matériel informatique ou de progiciels commerciaux ; certains constructeurs informatiques pour qui cette stratégie de *bundling* constitue un revirement par rapport à l'évolution passée de facturation séparée (*unbundling*), puis de désengagement partiel des activités de logiciels et de services.

Le développement de ces prestations fait partie d'un mouvement plus général que l'on observe dans de nombreux secteurs économiques, qui est la recherche de "solutions globales"

¹¹² Eurostaf estime que "le service lié à la vente de produit logiciel" représente "un nouveau segment en expansion pour les éditeurs de logiciels" (1996, p.131).

¹¹³ Par exemple Microsoft a développé une activité d'édition de manuels d'utilisation de ses progiciels, dont un des débouchés est représenté par les utilisateurs, qui ayant copié illégalement le progiciel ne possède pas sa documentation.

¹¹⁴ Dans le cas des banques, le coût d'interfaçage et d'adaptation du progiciel à l'environnement particulier de chaque entreprise rend celui-ci environ deux fois plus cher que son prix d'achat (Le Monde Informatique, 23 mars 1995). Une étude d'Inférence estime même que l'achat du progiciel ne représente que 1/8 du coût réel total en intégrant le paramétrage, l'installation, les développements spécifiques, l'interfaçage et la formation (Eurostaf, 1996, p.68).

visant à répondre aux besoins spécifiques des utilisateurs. L'adéquation de la "solution" aux attentes des utilisateurs dépend étroitement de la qualité de la relation de service qui se constitue entre le producteur et les utilisateurs. En effet, le problème à résoudre n'est pas en général clairement identifié sous toutes ses dimensions préalablement à la réalisation de la prestation, mais se construit progressivement au cours de celle-ci, et nécessite pour être résolu de façon satisfaisante la participation active (dans des proportions variables) de l'entreprise utilisatrice (coproduction).

Serge Bouchy appelle "*ingénierie des systèmes d'information*" cette " nouvelle approche des produits-services et des types de prestations de la profession où le sens du service, celui de la méthode, de l'économie et de l'innovation sont entremêlés et fédérés au bénéfice des entreprises qui découvrent une nouvelle expression et une nouvelle compréhension du métier" (1994, p. 22). Il précise que "ce concept nouveau touche aussi bien à la qualité exigée des produits qu'aux procédés et aux comportements nécessaires pour les obtenir" et "qu'à cette nouvelle vision doivent être associés des formes contractuelles novatrices, des rapports avec les clients plus impliquants, et une solidité de la société de service apte à accompagner durablement la démarche du client" (idem). Ces nouvelles formes contractuelles tendent à intégrer de plus en plus des engagements de résultats, par exemple une facturation qui peut être basé sur le temps de panne des systèmes, de préférence au temps passé pour les développer.

Le premier principe de cette démarche d'ingénierie qui "nécessite des itérations et des représentations progressives pendant tout le cycle de développement" (Serge Bouchy, 1994, p. 183) est "l'implication des utilisateurs qui doit être favorisée afin que leur dialogue avec les spécialistes de l'ingénierie soit une chance de réussite du projet conduit en commun" (idem, p. 45). Ceci rejoint les conclusions d'analyses plus générales de l'informatisation, qui ont mis en évidence les désillusions nées des tentatives de se substituer totalement aux acteurs individuels et collectifs (Francis Pavé, 1989). Pour ce faire, de plus en plus de SSII recrutent des informaticiens aux compétences plus étendues (pas seulement les subtilités des techniques informatiques, mais également des capacités de communication, d'ouverture, de compréhension des problèmes des clients), voire dans certains cas des spécialistes des

sciences sociales extérieurs à la sphère informatique (Faïz Gallouj, 1994, p. 204)¹¹⁵. Le développement des interactions avec les clients s'explique également par la compréhension qu'il ne représente pas seulement une amélioration de la prestation fournie dont bénéficierait exclusivement le client, mais qu'il peut également améliorer les compétences du prestataire, voire être à l'origine d'innovations "coproduites".

Selon Serge Bouchy, "le client d'aujourd'hui recherche un partenaire compétitif qui lui apporte la sécurité et la garantie du résultat" (1994 p. 313). Des relations de confiance peuvent s'établir grâce au développement du partenariat avec des objectifs en commun, des cultures à partager, des enjeux équitables (idem, p. 299-302) et grâce à l'existence de réseaux englobant les services informatiques internes du client et des prestataires externes. A défaut, la certification du prestataire, avec l'existence de normes qui imposent la conformité à des méthodes et procédures de réalisation du logiciel spécifiées et la certification du personnel utilisé par un système de qualifications professionnelles, peut permettre d'atténuer les incertitudes sur la prestation.

b - ...mais également à partir de logiciels libres

Des prestations équivalentes peuvent également être proposées à partir de logiciels libres. Dans la période récente, des sociétés se sont développées très rapidement sur ces activités avec des particularités dues à la nature libre du logiciel sur lequel s'appuient leurs offres.

Le premier type de sociétés est ce qui est appelé "éditeur" ou "distributeur" de logiciels libres. Certes les logiciels libres sont accessibles gratuitement (cf. supra), mais dans la mesure où un logiciel ne comprend aucun service lié, ne serait-ce qu'une simple garantie de bon fonctionnement, des sociétés se sont créées qui commercialisent un *package* constitué du logiciel sous une forme plus facile à installer (support CD plutôt que téléchargement, programme d'installation) et de services limités (aide en ligne, documentation, support technique éventuel). Il existe de nombreuses sociétés qui réalisent cette activité (Mandrake, Sendmail Inc, SuSE, Caldera, Slackware, Debian) la plus connue pour Linux étant la Red Hat.

¹¹⁵ Serge Bouchy estime que pour "faire exprimer les facteurs qualité le plus tôt possible par les utilisateurs", dégager " des facteurs qui ont une utilité et un intérêt réel pour le projet" et parvenir à des "exigences compatibles entre elles", il est nécessaire "de faire appel à une équipe de conception pluridisciplinaire" permettant "un travail en commun de tous les acteurs sur cette expression de la qualité" (1994, p. 189).

Son président et fondateur, Bob Young, estime qu'elle s'est imposée en proposant "un package plus simple à installer et à utiliser, et mieux documenté (...), le but de Red Hat [étant] de créer une marque, un label comme Heinz pour le ketchup ou Evian pour l'eau minérale" c'est dire "un nom, une technologie et une image uniques" (entretien avec Jean-Paul Smets-Solanes et Benoît Faucon, 1999, p. 97). Ce modèle économique n'est pas très éloigné de celui de Network Associates, qui est devenue la dixième entreprise de logiciels en termes de chiffre d'affaires, alors même que la plupart de ses progiciels (des logiciels anti-virus notamment) sont offerts gratuitement en téléchargement, l'entreprise faisant payer uniquement les mises à jour et le support technique (Carl Shapiro, Hal R. Varian, 1999, p. 86)¹¹⁶.

Il existe également des sociétés qui se sont spécialisées dans la vente de matériels intégrant des logiciels libres installés pour remplir des fonctions spécifiques comme des serveurs de réseau, par exemple Cobalt Micro aux Etats-Unis ou Mandala International en France. Ce type de solutions est proposé de plus en plus par des constructeurs informatiques traditionnels (Dell, IBM...).

Enfin il existe des *Value Added Reseller* (VAR) dont le rôle consiste à adapter une solution libre à une problématique dans une société donnée, puis à la déployer et à en assurer la maintenance, les travaux de développement et d'installation étant facturés au client. Cette activité est souvent menée en complément de celle d'éditeurs de logiciels libres, avec comme forme intermédiaire la vente de conseil, de support technique, de la formation et de l'intervention sur site. Elle est, pour le moment, effectuée essentiellement par des sociétés spécialisées sur les logiciels libres¹¹⁷. Aux Etats-Unis, VA Linux propose des services autour de Linux depuis 1993, mais c'est surtout à partir de 1997 que les logiciels libres commencent à être connus en dehors d'une sphère spécialisée et apparaissent comme des outils performants, surtout en tant que support des nouveaux services qui se développent autour d'Internet (Nicolas Jullien, 1999, p. 65) : on peut citer ESC Informatique qui travaille surtout avec le monde la recherche, Easter Eggs qui propose une solution simplifiée de commerce

¹¹⁶ Il faut toutefois souligner une différence, qui peut être importante, avec les éditeurs de logiciels libres : en étant seule à connaître le code source du produit principal Network Associates possède un avantage compétitif important sur d'éventuels concurrents qui voudraient se positionner sur le marché de ces prestations complémentaires.

¹¹⁷ Cependant le succès de certains logiciels libres peut faire que cette activité intéresse de plus en plus des SSII "traditionnelles". Par exemple, le succès d'Alcove, société française créée en 1996, lui a valu de signer récemment un accord de partenariat avec la SSII SQL Ingénierie.

électronique développée sous licence GPL (EEShop), Aurora qui réalise des sites Web sous Linux, Alcove qui s'est spécialisée dans la fourniture de services de haut niveau dans le domaine de l'informatique libre¹¹⁸. Ces sociétés sont de petite taille mais en croissance rapide (Alcove qui se considère comme le leader européen dans son domaine compte 25 salariés et prévoit de doubler ses effectifs au cours de l'année 2000).

En effet, des prestations réalisées à partir de logiciels libres présentent des avantages. La libre disponibilité du code source de ces logiciels rend ces sociétés complètement indépendantes des producteurs du logiciel, notamment dans ses possibilités d'évolution, ce qui explique une qualité et une variété fréquemment supérieures des services fournis. Surtout, les solutions basées sur des logiciels libres sont, à quantité de services comparables, beaucoup moins chères que les solutions basées sur des logiciels privés, puisqu'il n'existe aucune obligation technique ou économique pour que les sociétés qui commercialisent des services liés à des logiciels libres contribuent au code source du logiciel libre sur lequel sont basés leurs services¹¹⁹, ou paient des redevances pour le développement de ce code. Ceci peut contribuer à étendre l'accès à ce type de prestations - initialement limitées aux grandes entreprises et aux administrations - à des PME.

2 - Des logiciels sur mesure à partir de composants standardisés

a - Des potentialités importantes...

Dans cette seconde modalité du monde production flexible, le caractère "sur mesure" de la production s'étend au logiciel lui-même. Ce qui différencie cette forme de production de celle à l'œuvre dans le monde interpersonnel, c'est que les logiciels sont construits en réutilisant des modules logiciels déjà développés et testés. Il peut s'agir de l'utilisation de

¹¹⁸ Alcove a par exemple réalisé sous Linux, un serveur de terminaux codes-barres en communication radio et un serveur de télécopie avec interface Web. Elle compte déjà plus de 500 clients parmi lesquels des grands groupes français (l'Oréal, les Trois Suisses, Rhône-Poulenc).

¹¹⁹ Certes, pour le moment, la plupart de ces sociétés contribuent à des degrés divers au développement du logiciel libre ; on peut expliquer cette situation par une certaine proximité culturelle des acteurs présents initialement sur ce type de marché, par une stratégie marketing d'image de marque des sociétés concernées vis à vis d'utilisateurs experts sensibilisés aux enjeux des logiciels libres, par une volonté de développer des produits complémentaires à leur activité, et par une compétence supérieure dans les services liés que procurerait la participation au développement du code source des logiciels libres par rapport à une simple connaissance de celui-ci. Mais d'une part, il n'est nullement évident que cette situation se pérennise (cf. chapitre VIII) et surtout chaque société ne participe qu'à une petite partie des efforts de développement, vu le modèle de développement coopératif des logiciels libres.

plusieurs progiciels qui sont utilisés en tant que composants dans des systèmes plus vastes (Frederick P. Brooks, 1996, p. 248), ou de "l'assemblage" de composants logiciels disponibles "sur étagère" (certains parlent de "glu" logicielle pour décrire cette activité).

Cette tendance représente une étape supplémentaire dans l'évolution de l'informatique qui s'est toujours opérée par l'ajout de nouvelle "couche" entre l'ordinateur et l'utilisateur permettant d'éloigner la programmation de la machine et de la rapprocher des utilisateurs¹²⁰. Frederick P. Brooks compare la réutilisation à l'emploi d'un langage d'un plus haut niveau¹²¹ dont la taille du vocabulaire de programmation a radicalement augmenté (1996, p. 194). Ce qui permet cette évolution c'est l'arrivée à maturité des technologies "objets" qui autorisent une réutilisation facile des classes associée à une personnalisation grâce aux mécanismes de l'héritage (cf. chapitre III)¹²². Elle peut se combiner avec l'automatisation de certaines phases du processus de production des logiciels comme la documentation, la génération de code, les tests, grâce à des ateliers de génie logiciel : Luc Rubiello nomme *designware* l'activité qui consiste à développer avec des outils de génie logiciel et à partir de modèles conceptuels génériques stockés dans un dictionnaire central, des applications spécifiques (1997, p. 160-161).

Cette forme de production correspond dans l'économie du logiciel à la "convergence vers une production modulaire sur processus standardisés" (Anne Mayère, 1990) ou "sur mesure de masse" (Albert Bressand et Kalypso Nicolaidis, 1988) qui permettrait de surmonter la contradiction entre les besoins des producteurs d'une standardisation croissante apte à générer des économies d'échelle et les besoins des utilisateurs d'une individualisation du

¹²⁰ Frederick P. Brooks fait remarquer que cette activité, parfois appelée métaprogrammation, n'est pas sans rappeler la création de langages d'application entiers à partir de macro-instructions de langage assembleur, réalisée dans les années soixante (1996, p. 246).

¹²¹ En informatique, on oppose les langages de bas niveaux proches de la machine (le langage de plus bas niveau est le "langage machine" qui se limite au jeu d'instructions du processeur) mais difficilement utilisables par les humains, aux langages de hauts niveaux, plus riches et plus facilement utilisables, avec lesquels on produit du code qui est ensuite traduit automatiquement en langage machine par un compilateur ou un interpréteur.

¹²² Evidemment l'existence de langages de programmation possédant un vocabulaire plus étendu constitué d'une multitude d'objets, dont il faut connaître la syntaxe (les interfaces externes) et la sémantique (le comportement fonctionnel détaillé), nécessite un apprentissage initial plus important. Actuellement, on trouve des bibliothèques de classes de plus de 3000 membres, avec beaucoup d'objets pour lesquels on doit spécifier dix à vingt paramètres et options (Frederick P. Brooks, 1996, p. 194). Toutefois Frederick P. Brooks estime qu'il ne s'agit pas d'un obstacle insurmontable : "les gens utilisent couramment un vocabulaire de 10 000 mots dans leur langue maternelle, voire bien plus s'ils ont reçu une bonne éducation" (idem).

produit par rapport à leur spécificité. Si dans le monde industriel, cette évolution est permise par le passage d'équipements rigides à des équipements flexibles, dans l'économie du logiciel ce sont des investissements essentiellement immatériels qui deviennent flexibles.

La production de logiciels sur mesure à partir de composants standardisés est riche de potentialités par rapport aux questions critiques de la production des logiciels. Comparativement à un logiciel développé complètement sur mesure, la *réutilisabilité* permet une *productivité* bien supérieure, "l'emploi de composants interchangeables et réutilisables [permettant de] s'attaquer à l'essence conceptuelle du problème" (Frederick P. Brooks, 1996, p. 182). Tout en soulignant que la réutilisation ne se limite pas au code mais peut concerner douze composants réutilisables (architecture, cahier des charges, plans, estimations, conception et spécification, interfaces, données, écrans et éléments d'écrans, code source, documents d'utilisation, plans de tests, jeux de tests), Capers Jones estime à 65 % l'amélioration de la productivité et à 50 % l'amélioration des délais (1998, p. 35).

Concernant la *fiabilité*, celle-ci peut être grandement améliorée par le fait que les différents composants ont déjà été testés dans de nombreuses situations, les problèmes devant se limiter à ceux qui résultent de leur intégration. Capers Jones évalue à 85 % la réduction des défauts résultant de la réutilisation (1998, p. 35).

Ce modèle de développement peut également permettre une meilleure adéquation aux besoins des utilisateurs. Dans le cas d'un logiciel, "l'individualisation" n'est pas une dimension supplémentaire rajoutée à un produit à la valeur d'usage bien établie comme pour les produits manufacturés, mais une dimension essentielle de l'utilité du logiciel. Ce phénomène prend d'autant plus d'importance que les fonctions qui ont été les premières informatisées sont celles qui correspondaient à des besoins standards, qui pouvaient être couverts par des progiciels, éventuellement en les paramétrant (paye, comptabilité, gestion des stocks...). De plus en plus l'informatisation concerne des aspects particuliers à chaque organisation, et "les choix des firmes doivent garantir le maintien de leurs spécificités et de leur culture" (Claire Charbit, Jean-Benoît Zimmermann, 1997, p. 22). Dans une situation où une conception organisationnelle originale peut constituer un avantage compétitif, il faut tenter de lui faire correspondre une solution technique adéquate : "il ne s'agit donc pas d'assujettir l'organisation de l'entreprise aux caractéristiques du système d'information, mais plutôt de procéder à l'inverse" (idem, p. 21).

Ce type de développement peut se combiner avec l'utilisation de méthodes centrées sur l'utilisateur (prototypage rapide, méthodes de développement incrémentales et conception participative) qui permettent à la différence des méthodes traditionnelles, une amélioration très rapide des applications développées (Thomas K. Landauer, 1997, p. 8). Pour tirer véritablement parti des potentialités des conceptions orientées "objet", il est nécessaire (*et possible*) que les utilisateurs participent activement et continûment (Eurostaf, 1996, p.79). Selon Serge Bouchy l'utilisation des langages de quatrième génération, "logiciels outils grâce auxquels le réemploi multi-domaines applicatifs devient une réalité quotidienne (...), permet d'atteindre des objectifs quantitatifs en termes de charge de production et de délai de mise à disposition d'applications nouvelles, et qualitatifs en termes de relation entre les concepteurs d'applications et/ou les moyens technologiques d'une part, les usagers finals d'autre part" (1994, p. 183-184). Il ajoute que "de plus, le rôle des différents acteurs (utilisateurs, concepteurs, informaticiens) se trouve rééquilibré au bénéfice des usagers finals qui prennent en charge une part plus déterminante de la conception et de la réalisation de leur propre système d'information automatisée" (Serge Bouchy, 1994, p. 163). Un exemple intéressant est celui d'une des principales compagnies d'assurance française, dont la description du nouveau système mis en œuvre est effectuée ainsi par le directeur informatique : "Ce qu'on a fait sur la base de notre architecture, c'est un catalogue dans lequel les actuaires disent "tiens je vais inventer un contrat incendie consécutif à des orages". Et ils prennent dans le catalogue des choses, des paramètres et ils les assemblent et cela génère le produit (...). Au lieu d'un schéma ancien : conception, demande à l'informatique, cahier des charges, réalisation, mise en œuvre, schéma très long et où on ne peut pas faire de jetable ni de prototype, on a un outil mis à la disposition uniquement des actuaires qui peuvent générer eux-mêmes le produit dont ils ont besoin (...). Cela a un rapport étroit avec le "développement orienté objets", qui touche tout le monde : c'est l'utilisateur qui manipule ces objets, et je pense d'ailleurs qu'un jour ou l'autre il y aura besoin de moins d'informaticiens dans la mesure où les utilisateurs manipuleront directement des "objets" pour concevoir des produits, les gérer" (entretien réalisé par Faïz Gallouj, 1997, p. 60). Ce qui rend également possible cette évolution, c'est l'amélioration des compétences informatiques dans le maniement d'outils plus "conviviaux" de la part des non-informaticiens, grâce notamment au développement de la microinformatique et à l'utilisation croissante des réseaux.

Enfin, le dernier avantage d'un logiciel sur mesure par rapport à un progiciel, est de ne pas comporter des fonctionnalités non utilisées - la tendance des producteurs de progiciels

étant de fournir un maximum de fonctionnalités pour couvrir avec un produit dont les coûts sont presque uniquement fixes un marché le plus étendu possible – qui peuvent de plus augmenter la complexité d'utilisation et dégrader la fiabilité.

b - ...mais des difficultés pour se développer

Malgré ces avantages indéniables, la réutilisation des composants logiciels a pourtant du mal à se développer.

Au niveau interne à une organisation, il faut tenir compte du fait qu'une pratique de réutilisation demande des investissements initiaux importants. C'est qu'en effet, pour que du code soit réutilisable, il faut que le "software [soit] écrit depuis le début avec l'intention de faciliter sa réutilisation" (OTA, 1992, p. 154). Il est nécessaire : "d'identifier les caractéristiques des composants des logiciels, qui facilitent le réemploi ; de définir les techniques permettant de traduire un composant ne pouvant être réutilisé que marginalement en un composant largement réutilisable ; de développer des systèmes de classification et de repérage des composants, en vue de faciliter leur recherche dans une base de données, en cas de besoin" (idem, p. 155). En particulier, une réutilisation réussie exige un contrôle de la qualité qui approche ou atteint le niveau du zéro défaut (Capers Jones, 1998, p. 35), ce qui, selon Frederick P. Brooks, rend les composants réutilisables trois fois plus coûteux à développer que les composants à usage unique. Si cet investissement initial pourrait rapidement être amorti ¹²³, il n'est pas toujours effectué à cause de la crainte que les composants développés soient peu utilisés en raison de l'opinion fréquente chez les programmeurs, même si le plus souvent non fondée, qu'il faudra finalement moins d'effort pour développer les composants dont il a besoin. Il faut également prendre en compte dans le cas des SSII, leur mode de fonctionnement économique basé sur des activités à cycle très court, d'où une certaine réticence à investir, les activités réalisées devant donner lieu à facturation immédiate.

Il faut également envisager le cas d'une réutilisation qui ne se limite pas à l'intérieur de l'organisation qui a produit les composants. C'est bien évidemment dans ce cas que les potentialités sont les plus intéressantes. Le développement de marchés de composants

¹²³ Capers Jones estime même que "la réutilisabilité offre la plus grande valeur ajoutée et le meilleur retour sur investissement" (1998, p. 23).

logiciels, régulièrement annoncé, est dans la plupart des cas resté à un stade embryonnaire¹²⁴. En effet, de façon encore plus impérative que pour les logiciels, la réutilisation de composants nécessite de disposer du code-source (Eric Raymond, 1998), notamment pour éliminer les effets inattendus résultant de l'assemblage des composants logiciels ; mais dans ce cas, il devient quasi impossible pour le producteur de facturer la fourniture de ces composants¹²⁵, sauf en la liant à la réalisation d'autres prestations. Une récente étude de l'OCDE consacrée aux perspectives des technologies de l'information estimait que " les débouchés pour les objets sont encore relativement étroits, et la plupart des développeurs travaillent avec des objets fournis par leur environnement de développement ou qu'ils ont construit eux-mêmes. Les techniques de POO (Programmation Orientée Objet) ont aidé à améliorer la gestion des grands projets de développement, en fait surtout parce qu'elles facilitent une division du travail bien définie au sein d'une société, et non pas parce qu'elles facilitent des relations plus élaborées entre entreprises" (1997 B, p. 184). Une solution pourrait être le développement de ces composants sous forme de composants logiciels libres, mais pose la question des incitations pour produire de tels composants (cf. chapitre VIII).

Après avoir développé les fondements socio-économiques des mondes de production et exposé une typologie des mondes de production basée sur les travaux de Robert Salais et Michael Storper, nous avons appliqué cette typologie à l'économie du logiciel. Ce cadre théorique nous a permis d'interpréter la coexistence dans l'économie du logiciel de formes très différenciées de l'activité économique, et d'analyser par rapport aux questions "critiques" les avantages mais aussi les inconvénients de chacun des mondes de production, très différents selon les cas : difficultés à concilier productivité et fiabilité élevées pour la production de logiciels sur-mesure (monde interpersonnel), forte productivité mais qualité souvent

¹²⁴ Frederick P. Brooks parle d'un "hypothétique décollage du marché des classes en C++" (1996, p. 247).

¹²⁵ Les questions juridiques sont de toute façon complexes pour un logiciel constitué de composants produits par des acteurs économiques divers. Empêcher une appropriation illégale est déjà difficile pour un progiciel complet, et semble quasiment impossible pour un composant logiciel. Si l'on peut faire la preuve que quelqu'un utilise un progiciel donné, il est par contre moins évident de prouver qu'un logiciel contient un composant logiciel donné, surtout quand on ne dispose pas du code-source du logiciel en question.

décevante et limitation à des besoins standards pour les logiciels commerciaux (monde fordiste), originalité et efficacité apportées par des logiciels libres sur certains segments mais difficultés de généralisation d'un tel modèle (monde de la création), potentialités pour concilier productivité et adaptation aux besoins des utilisateurs du monde de production flexible mais difficultés de développement d'un tel monde, surtout dans sa composante la plus prometteuse (la production de logiciels sur mesure à partir de composants standards). Il nous reste à analyser les évolutions dynamiques, dont l'existence de ces quatre mondes est le produit, et à esquisser quelques perspectives d'évolution.

Chapitre VIII

COMPLEMENTARITE ET DYNAMIQUE

DES "MONDES DE PRODUCTION" DU LOGICIEL

Une explication fréquente de la pluralité des formes de production dans l'économie du logiciel est la jeunesse de cette activité. Par exemple, Christian Genthon et Denis Phan constatant que "l'industrie du logiciel est encore dans sa phase de démarrage" soulignent que les "produits, modes de production et modes de distribution ne sont pas encore totalement stabilisés" (1999, p. 187). Si ce constat contient une part indéniable de vérité, il ne doit pas induire l'idée qu'une économie du logiciel parvenue à maturité se caractériserait par une domination sans partage d'un seul monde de production, qui reléguerait les autres mondes au statut de survivances historiques dépassées, de manière analogue au triomphe de la production industrielle sur la production artisanale.

Ce qui permet au contraire de penser que la coexistence de différents mondes de production est un phénomène durable, c'est qu'*il existe entre les différents mondes de production des logiciels, d'importantes relations de complémentarité*. Tout d'abord les différents mondes de production des logiciels correspondent à des types de produits différents, produits qui sont eux-mêmes souvent complémentaires¹. Par exemple, il n'existe aucune raison pour que le modèle économique qui s'impose pour développer un site Web de commerce électronique, pour lequel la nécessité d'une adaptation fine aux caractéristiques de l'entreprise est primordiale, soit le même que celui qui permet de produire des systèmes d'exploitation, pour lesquels la standardisation est indispensable. Or la diversité des logiciels a tendance à s'accroître avec le développement de l'informatisation. De plus pour un même type de produit, les utilisateurs sont très divers et accordent une importance différente aux multiples caractéristiques d'un logiciel, dont on a vu qu'elles étaient dépendantes des mondes de production. Si l'on prend l'exemple des systèmes d'exploitation, un utilisateur expert très

¹ Cette complémentarité des logiciels peut concerner leur utilisation mais aussi leur production, un logiciel étant en grande partie produit à l'aide d'autres logiciels.

sensibilisé aux questions de performance et de fiabilité pourra préférer un logiciel libre comme Linux, alors qu'un simple usager, essentiellement motivé par la facilité d'utilisation et d'installation (surtout pour les périphériques de plus en plus divers et nombreux des ordinateurs), se tournera généralement vers le système le plus diffusé (Windows).

Un deuxième aspect de la complémentarité entre les mondes de production des logiciels, est que dans certains cas, le produit d'un monde peut servir d'input à la production d'un autre monde. C'est notamment le cas pour le monde de production flexible où des prestations sur mesure peuvent être construites à partir de logiciels commerciaux ou de logiciels libres. Ces complémentarités ne sont pas que techniques et peuvent être à la base d'un renforcement dynamique simultané des deux mondes de production correspondants : par exemple, l'existence de sociétés proposant des services autour de Linux renforce la diffusion de ce logiciel, ce qui en retour incite à proposer de nouveaux services².

Un troisième aspect concerne les incitations que peut représenter le développement d'un monde de production avec ses caractéristiques spécifiques, sur la production d'un autre monde. Nous avons souligné que les qualités parfois jugées insuffisantes des logiciels commerciaux avaient pu motiver la création de logiciels libres. Réciproquement, le succès acquis par certains d'entre eux entraîne des réactions des producteurs du monde fordiste, soumis à une nouvelle concurrence. De façon plus générale pour les produits et services d'informations, Anne Mayère identifie trois modes d'échange différents (échanges non marchands, échanges de marchandises et échanges de services), mais effectue cette précision importante : "une erreur d'interprétation serait de les concevoir comme antinomiques et d'envisager leur évolution relative comme un jeu à somme nulle. Le développement des marchés de l'information repose au contraire sur la dynamique combinée de ces trois composantes, et notamment sur des passages et combinaisons constantes entre ce qui est de l'ordre du marchand et ce qui est de l'ordre du troc. C'est en effet dans cette dynamique du gratuit et du payant que se régulent l'identification des valeurs d'usage et marchande de l'information et leur traduction réciproque, opérations toujours difficiles et toujours recommencées" (1997, p. 142).

² Ce type de cercle "vertueux" ne se limite pas aux solutions constituées autour de logiciel libre. On peut observer le même phénomène (développement interdépendant d'un logiciel et des sociétés proposant des services complémentaires) avec SAP pour les logiciels de gestion d'entreprise ou avec Oracle pour les logiciels de gestion de bases de données.

En conséquence, la présence de plusieurs mondes de production dans l'économie du logiciel nous paraît devoir se perpétuer. Toutefois cette coexistence ne doit pas masquer les profondes évolutions de ce secteur, qui se concrétisent notamment par des changements dans l'importance respective de chacun de ces mondes. Il se produit entre les différents mondes de production des évolutions dynamiques que nous allons analyser en identifiant les principales trajectoires à l'œuvre.

Cette analyse, qui a un caractère plus hypothétique, sera moins développée que celle du chapitre précédent. Si nous avons pu repérer de façon inductive les trajectoires les plus fréquentes et proposer quelques éléments d'explication, la construction d'un cadre théorique complet et adapté, qui intégrerait notamment les apports des théories évolutionnistes, reste à effectuer. De plus comme certains phénomènes sont très récents (le développement du monde de la création, certaines modalités du monde de production flexible), ils ne permettent pas de prendre le recul nécessaire pour pouvoir valider les hypothèses émises, ce qui nécessiterait des investigations supplémentaires.

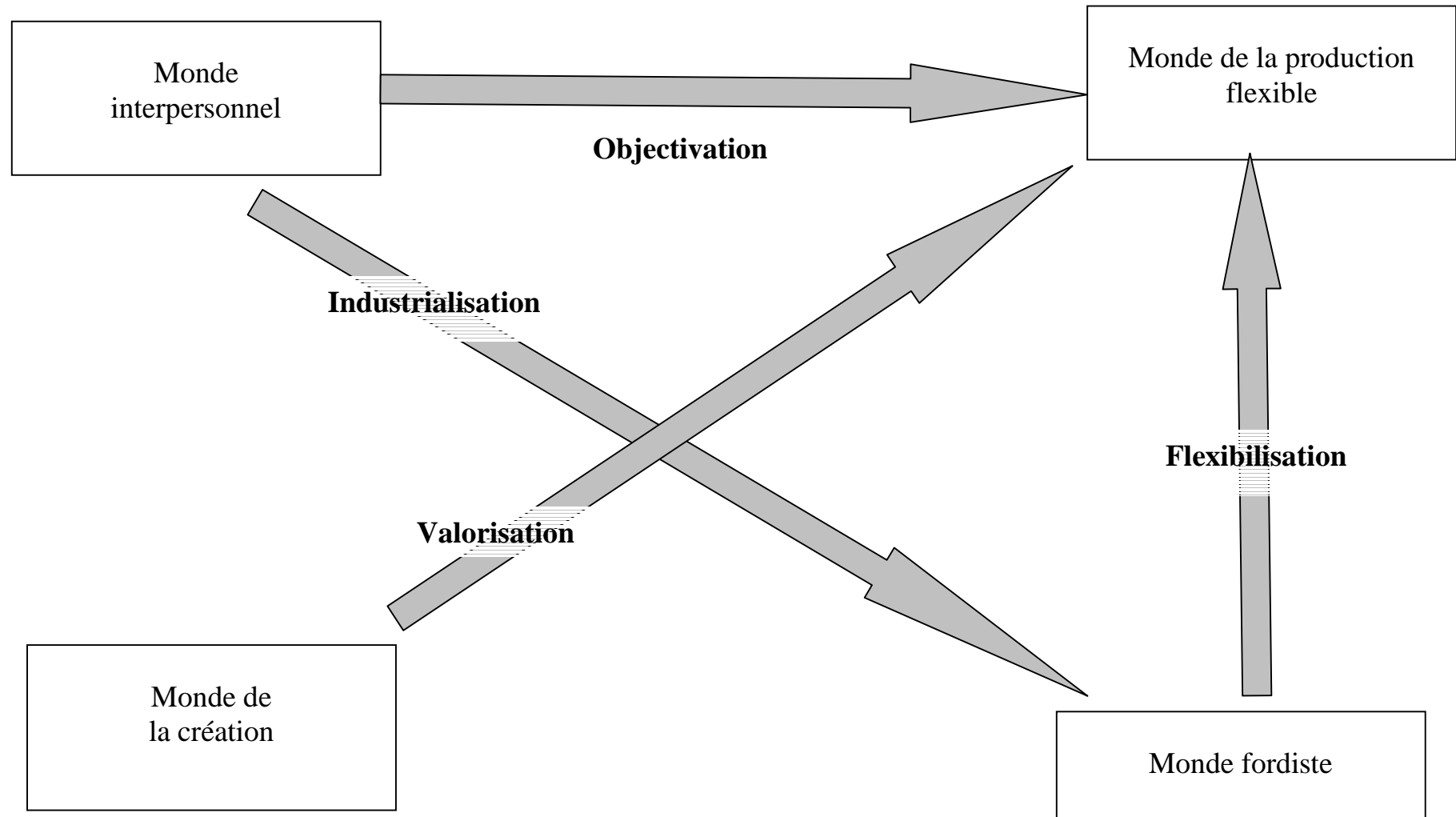
Par contre, l'intérêt d'une telle analyse, même partielle, est de montrer comment certaines dynamiques ont déjà affecté différemment les trois grandes zones de production des logiciels que sont les Etats-Unis, le Japon et l'Europe (section II) et de mettre en évidence les perspectives d'évolution les plus souhaitables – mais pas nécessairement les plus probables – pour l'ensemble de l'économie du logiciel, ce qui peut motiver des actions volontaristes (section III). Auparavant les quatre trajectoires principales d'évolution sont analysées dans une première section.

Section I - Les évolutions dynamiques de l'économie du logiciel : quatre trajectoires principales

Toutes les évolutions que connaît l'économie du logiciel ne se traduisent pas par un changement de monde de production. Il existe en permanence de multiples démarches pour améliorer l'efficacité de la production des logiciels que l'on peut qualifier de rationalisations "mineures", dans la mesure où elles ne modifient pas en profondeur le cadre conventionnel qui est à la base d'un monde de production. Mais le plus important nous semble être constitué par les rationalisations "majeures" qui sont marquées par *un déplacement d'une production effectuée dans un monde de production vers un autre monde de production*. On peut distinguer, parmi les multiples possibilités théoriques³ existantes, quatre trajectoires qui résument les principaux changements observés dans l'économie du logiciel (cf. schéma XII) :

³ En théorie, parce qu'en pratique beaucoup de trajectoires sont impossibles.

Schéma XII : Les évolutions dynamiques de l'économie du logiciel



Avant d'analyser ces quatre trajectoires, quelques précisions sont nécessaires, notamment pour éviter des erreurs d'interprétations que pourrait induire leur représentation schématique. Si le passage d'un monde de production à un autre peut s'effectuer par la transformation de l'activité économique d'une entreprise, cette évolution n'est pas la plus fréquente. En effet, une telle transformation implique un changement radical de l'identité de l'entreprise concernée, dans ses objectifs, dans le type de produit qu'elle réalise, dans son organisation, dans ses rapports avec les utilisateurs, changement dont l'ampleur le rend difficilement acceptable. Il s'agit donc plus souvent de trajectoires de produits que d'entreprises, le passage d'un monde de production à un autre, s'effectuant par la croissance de l'activité des entreprises et l'apparition de nouvelles entreprises dans le monde de production de destination, accompagné éventuellement de la chute de l'activité, voire de la disparition d'entreprises, dans le monde d'origine. D'autre part, l'existence d'une trajectoire partant d'un monde de production ne condamne pas celui-ci à un déclin irrémédiable. Outre l'existence de relations qui peuvent être de complémentarité⁴ plus que de substitution entre les mondes (cf. supra), l'apparition permanente de nouveaux segments d'activité peut susciter de nouveaux développements du monde d'origine : par exemple, en même temps que certains logiciels sur mesure sont remplacés par des progiciels, de nouveaux domaines d'applications apparaissent qui au départ peuvent être développés sur mesure avant qu'une généralisation éventuelle voit l'apparition de progiciels.

A - INDUSTRIALISATION

La première de ces évolutions a été "l'industrialisation" de l'économie du logiciel, dont l'objectif était l'amélioration de la productivité dans la production des logiciels, par le passage de produits sur mesure à des produits standards (progiciels). La notion d'"industrialisation" est ambiguë et, comme nous le verrons, peut recouvrir des phénomènes assez différents. Nous avons néanmoins choisi ce terme, car d'une part c'est celui qui est très largement utilisé pour désigner le nécessaire passage d'un "stade artisanal" à un "stade industriel" dans la production des logiciels (Patrick Jaulent 1992, p. 14)⁵, et d'autre part, il est utilisé pour désigner dans les

⁴ Dans ce cas, la croissance d'un monde de production peut servir de soubassement à l'essor d'un autre monde.

⁵ La nécessité d'une "industrialisation" de la production des logiciels revient de manière récurrente dans la littérature spécialisée. Pour se limiter à deux exemples, Eurostaf estime que "les SSII françaises passent d'une

services une évolution assez proche de celle que représente la trajectoire *monde interpersonnel* → *monde fordiste* dans l'économie du logiciel. Par exemple, Faïz Gallouj décrit ainsi le processus d'industrialisation des services à partir d'une situation de "service pur" où les caractéristiques de service sont obtenues en mobilisant exclusivement des compétences, celles du prestataire combinées à celles du client (coproduction) : le processus d'industrialisation se manifeste dans un premier temps, par la disparition progressive du client en tant qu'acteur de la production (affaiblissement, puis disparition de la relation de coproduction) et la mise en œuvre de technologies immatérielles. Ce processus converge vers un service assimilable à un quasi-bien, caractérisé par le recours croissant aux technologies matérielles et au self-service et par la disparition de la mobilisation directe de compétences "au contact" (1997, p. 67).

Si l'industrialisation implique que "la production du service s'autonomise vis à vis du client et du prestataire" (Faïz Gallouj, 1994, p. 118), l'affaiblissement de la relation de service qui en résulte peut parfois coïncider avec une amélioration du service rendu : "il existe pour une entreprise de service d'autres moyens de rendre un service jugé supérieur que l'approfondissement des connexions avec ses clients : le caractère adapté du cadre technique, spatial ou architectural de la prestation, la mise à disposition d'outils techniques utilisés en self-service (...) la diversité des offres et des solutions proposées, le professionnalisme supérieur, la réduction des dysfonctionnements et des erreurs..." (Faïz Gallouj, 1997, p. 67).

Cependant, le plus souvent, la notion d'industrialisation est définie par analogie avec le "mode de production (organisation du travail et techniques) dominant dans la grande industrie de l'après-guerre" (Jean Gadrey, 1994 A) : elle "désigne un processus au cours duquel une catégorie d'organisation n'appartenant pas au monde industriel tend à se rapprocher de ce dernier, au moins sur certains plans jugés significatifs : concentration de moyens de production et de main d'œuvre, recours à des outillages et machines souvent perfectionnés pour leur époque et (...) capacité à produire en série des biens standardisés dans le cadre d'organisations mécanistes" ; elle "se réfère à des méthodes de gestion et de mesure des performances, souvent liées à la recherche prioritaire de gains de productivité et de réduction des coûts standard" (idem, p. 167, 168 et 171).

logique artisanale à une logique industrielle" (1996 A, p. 248) et Serge Bouchy explique que "l'ensemble de la démarche de génie logiciel vise à *industrialiser* la production des logiciels" (1994, p. 163).

C'est plutôt cette vision de l'industrialisation, basée sur une accentuation de la division du travail au sein de grands établissements⁶, des "efforts de formalisation des savoir-faire [mobilisant] des ressources intellectuelles et financières importantes" (Serge Bouchy, 1994, p. 21) et des investissements lourds dans des ateliers de génie logiciel intégrés (comme AD Cycle produit par IBM) qui était défendue par de nombreux auteurs. Par exemple Jean Bres évoquait une vision de l'industrialisation de l'économie du logiciel explicitement similaire à celle qu'avait connue le secteur automobile : "on peut faire un parallèle entre le point où en est aujourd'hui l'informatique, et celui où en était l'industrie automobile avant qu'Henry Ford invente la chaîne : avant l'invention de la chaîne, des mécanos géniaux fabriquaient des automobiles dans leur garage en les assemblant à la demande, et en faisant toutes les pièces une par une ; après Henry Ford, l'industrie automobile est passée à un stade industriel, où on utilise des machines outils, où la chaîne d'assemblage a permis d'enregistrer la connaissance dans une organisation et non plus dans la tête des gens. (...). On peut dire qu'il est indispensable que l'informatique réussisse la mutation qu'a réussie l'industrie automobile : passer à une approche beaucoup plus industrialisée, beaucoup plus standardisée ; de même que la chaîne représente à la fois des machines outils et une certaine organisation du travail, c'est avec à la fois des méthodes, et des outils, qui permettent de supporter ces méthodes, que l'informatique réussira à casser le cercle vicieux du manque de productivité" (1994, p. 6).

Cette forme d'industrialisation s'est plus particulièrement développée au Japon avec la constitution de véritables usines à logiciel, analysées par Michael Cusumano (1991). Pour certains auteurs c'est ce modèle qui devait s'imposer, ses difficultés d'implantation résultant de l'ampleur des investissements nécessaires (en matériel, outils logiciels et formation) avec des délais de recouvrement mal définis, et des difficultés à le faire accepter par les informaticiens. En effet, ceux-ci se trouvaient dans une position de forces, d'une part en raison de l'étendue de leurs compétences techniques, d'autre part en raison de la relative pénurie d'informaticiens qu'une amélioration de la productivité insuffisante par rapport à la croissance des besoins, contribuait à maintenir. Une étude de l'OCDE en 1991 estimait ainsi que le principal frein résidait dans l'existence de "rigidités sociales et institutionnelles face aux technologies et aux méthodes nouvelles" (1991 A, p. 27). De nombreux programmeurs considèrent le génie logiciel comme une menace en raison de l'obsolescence de leurs compétences, dans une

⁶ "Trop souvent encore, certaines tâches d'un projet ne peuvent être confiées qu'à un seul individu qui doit les mener du début à la fin " (Serge Bouchy, 1994, p. 146).

activité où il est plus long et plus coûteux de recycler le personnel en place que de former de nouvelles recrues (idem, p. 44). Ils ont de fortes réticences pour passer d'une situation où ils bénéficient d'une "large autonomie dans le travail" à une "organisation du travail plus formaliste, structurée et administrée" avec des possibilités "métrologiques de surveiller et noter le personnel" (idem, p. 29).

De toute évidence, c'est une autre forme d'industrialisation qui l'a emporté. Celle-ci a certes abouti à une production en grande série de biens standard mais effectuée par des entreprises aux caractéristiques organisationnelles distinctes de celles exposées précédemment, ce que nous avons appelé un "fordisme spécifique" (cf. chapitre VII). De ce fait, ce sont essentiellement de nouvelles entreprises avec des compétences, une organisation et une culture très différentes, qui vont s'imposer sur les marchés de progiciels, alors que les grandes entreprises de services informatiques et les grands constructeurs informatiques vont largement échouer malgré des tentatives répétées⁷. Cette forme d'industrialisation, qui s'est d'abord développée pour les logiciels pour micro-ordinateurs avant de s'étendre à d'autres domaines de l'économie de l'informatique, a ainsi été à la fois une cause majeure et une conséquence du développement de l'informatisation, avec un cercle vertueux classique de baisse des prix, d'augmentation de la diffusion et d'économies d'échelles. Si son extension peut se poursuivre dans certains segments où s'effectuent encore beaucoup de développement sur mesure, et si des possibilités existent que des domaines nouveaux soient couverts d'emblée par des progiciels, il n'en demeure pas moins que les progiciels ne peuvent répondre par définition qu'à des besoins standardisés. Or les problèmes standard ne correspondent qu'à une partie vraisemblablement décroissante des problèmes que des logiciels sont et seront amenés à résoudre ; ceux-ci sont de plus en plus complexes en se rapprochant de plus en plus des situations concrètes où une machine doit agir automatiquement ou aider l'utilisateur à prendre une décision. L'existence de besoins plus spécifiques explique l'attrait du monde de production flexible, qui permet d'obtenir une prestation sur mesure tout en minimisant les coûts, les délais de production et les défauts des logiciels produits dans le monde interpersonnel. Celui-ci peut être atteint à partir des trois autres mondes de production, ce qui correspond aux trois autres trajectoires mises en évidence.

⁷ Même s'il existe d'autres facteurs explicatifs, les échecs d'IBM, entreprise occidentale qui a le plus développé, dans le domaine du logiciel, les caractéristiques d'une "hiérarchie mécaniste industrielle", sont éloquentes : abandon du projet *Future System* sur lequel a travaillé plus de 2000 personnes dans les années soixante dix, échec d'*OS/2* qui a mobilisé un millier de programmeurs dans les années quatre-vingt.

B - FLEXIBILISATION

La première des ces trajectoires, que nous appelons *flexibilisation*, consiste à passer du monde fordiste au monde de la production flexible, par une "adaptation de plus en plus fine de l'offre à la demande [qui] implique un mouvement d'individualisation à partir de la production de masse" (Albert Bressand et Kalypso Nicolaidis, 1988, p. 149). Cette trajectoire, qui a fait l'objet de multiples analyses convergentes, est souvent privilégiée dans sa capacité supposée à surmonter la crise actuelle analysée comme étant la crise du fordisme. Le fordisme serait remis en cause dans son aptitude à continuer à générer de forts gains de productivité et dans sa capacité à répondre à des demandes plus volatiles et plus diversifiées. La production d'une gamme plus diversifiée de produits livrés en juste à temps, les techniques d'association (*bundling*) de biens et de services dont certains sont standardisés et les transformations qui les accompagnent permettraient d'obtenir simultanément une plus grande efficacité des producteurs et une meilleure adéquation aux besoins des utilisateurs.

Dans l'économie du logiciel, cette trajectoire se concrétise par la fourniture autour d'un progiciel, de services liés d'installation, de paramétrage, de formation, de maintenance, jusqu'à la réalisation de développements complémentaires sur mesure permettant d'individualiser le progiciel en fonction des besoins spécifiques de l'utilisateur. L'aboutissement de cette trajectoire réside dans la fourniture d'une solution globale et personnalisée intégrant des progiciels, du matériel et des services. Ces différentes activités, complémentaires de la production des progiciels, peuvent être réalisées soit par des sociétés spécialisées, qui se sont créées pour exercer cette activité ou qui se sont reconverties à partir d'activités en déclin comme le "service bureau", soit comme activité au départ secondaire de la part de producteurs de progiciels (Oracle par exemple) ou de constructeurs d'ordinateurs (IBM par exemple), ce que traduit la part croissante prise par les services dans leur activité. Les prestations plus ambitieuses, comme l'intégration de systèmes, sont fréquemment réalisées par des partenariats entre des entreprises complémentaires.

Cette trajectoire a été privilégiée dans les explications de la montée du monde de production flexible car elle est largement dominante dans le cas des biens industriels. Il existe toutefois d'autres possibilités pour parvenir à une production "sur mesure de masse" concernant des prestations logicielles.

C - OBJECTIVATION

La première de ces possibilités est représentée par une trajectoire aboutissant au monde de production flexible, à partir du monde de production interpersonnel. Celle-ci consiste, à partir de la production d'une prestation réalisée initialement complètement sur mesure, à appliquer des techniques de dissociation (*unbundling*) qui permettent d'extraire des composants et des méthodes suffisamment "impersonnels" pour pouvoir être réutilisés pour la production d'autres produits plus ou moins différenciés. Nous appelons *objectivation*⁸ cette trajectoire pour rendre compte de la transformation de stades d'un processus de production dédié au départ à un utilisateur particulier, en méthodes et composants standardisés, utilisés pour la production de produits destinés à des utilisateurs divers. Cette évolution permet de dégager des gains de productivité (en objectivant des méthodes réutilisables ou des composants qui sont facilement reproductibles) tout en maintenant une adaptation aux besoins particuliers du client-usager. C'est par exemple le cas d'une société de service qui développe un système expert en coproduction avec un client et qui réemploie la méthode élaborée et des modules logiciels pour proposer des systèmes expert à d'autres clients dans des domaines d'application différents.

Cette évolution correspond à une "rationalisation de type cognitif qui s'appuie sur la *typification* des cas, la *formalisation relative* des procédés (ou méthodes intellectuelles) de résolution de problèmes et le recours à un *répertoire de "routines"* que les individus peuvent avoir acquis à titre personnel mais qui existent souvent en tant que compétences collectivement diffusées dans et par l'organisation" (Jean Gadrey, 1994 A, p. 172). De ce point de vue, elle diminue la dépendance que M. Gibbs (1994) juge excessive envers les connaissances tacites des programmeurs, et qui serait responsable de la faiblesse des gains de productivité. Cette forme de rationalisation peut être qualifiée, en reprenant les termes de Jean Gadrey (1994 A) de "rationalisation professionnelle" - même si elle a une base technique plus importante que dans les "professions" analysées par la sociologie des professions - opposée à une "rationalisation industrielle" dont nous avons souligné la relative inefficacité dans la production des logiciels.

⁸ La notion d'objectivation a été initialement introduite par Faiz Gallouj (1994) pour désigner une forme particulière d'innovation dans les services.

L'existence de cette trajectoire *directe*, entre le monde de production interpersonnel et le monde de production flexible, constitue une certaine originalité de l'économie du logiciel, même si elle peut s'observer également dans certains services professionnels. En effet, dans l'industrie et dans la plupart des services, l'évolution consiste plutôt en un passage du monde interpersonnel au monde fordiste, puis du monde fordiste au monde de production flexible⁹. Par contre pour la production de logiciels, cette évolution peut s'effectuer directement¹⁰, y compris par la transformation de l'activité d'une entreprise¹¹, voire par la constitution d'une société indépendante à partir des services internes d'une entreprise (forme d'externalisation), qui va généraliser à un marché élargi les développements et les compétences provenant d'une activité au départ uniquement interne (exemple d'EDS par rapport à General Motors).

D - VALORISATION

La dernière trajectoire identifiée correspond au passage du monde de la création au monde de la production flexible. Nous l'appelons *valorisation* pour désigner la création d'activités marchandes à partir d'une production réalisée dans un cadre essentiellement non marchand. De façon générale, des formes de valorisation d'activités non marchandes existent couramment dans l'ensemble de l'économie, qu'il s'agisse de l'utilisation d'inventions par des entreprises existantes, ou du processus qui peut conduire un acteur du monde de la création, à partir de la valorisation marchande d'une de ces productions, à ne plus développer qu'une activité dérivée de cette invention en fonction des demandes du marché, en négligeant son activité créatrice d'origine, ou en la maintenant dans un rôle subordonné (création d'entreprise, filialisation d'un département de recherche d'une entreprise, création de structures ad hoc pour démarrer une production à partir d'une création aux perspectives commerciales

⁹ Ces trajectoires se succèdent certainement plus rapidement dans les services que dans l'industrie, ce qui n'est pas sans poser des problèmes : par exemple, le passage interpersonnel → fordiste se traduit fréquemment par la domination d'un travail largement prescrit par des services spécialisés parfois jusque dans les moindres détails, alors que le passage fordiste → flexible nécessite à l'inverse implication et initiatives des salariés.

¹⁰ Cette évolution directe correspond à celle que préconise Serge Bouchy : "en terme d'objectifs, il faut tout faire aujourd'hui pour que la profession de l'ingénierie informatique se normalise en "sautant" un stade de l'évolution industrielle (celui de la fabrication en usine, assurée par des individus), en passant très vite à l'ère des individus peu nombreux mais très efficaces, assistés par des outils robotisés au maximum. Il serait judicieux de voir la profession informatique utiliser pour son propre compte les outils de "conception assistée par ordinateur" qu'elle cherche tant à faire utiliser par les autres" (1994, p. 146).

¹¹ Dans ce cas, il ne s'agit plus seulement d'une trajectoire de produits mais d'une trajectoire d'entreprises.

prometteuses...)¹². Ce phénomène a été particulièrement marqué dans l'ensemble de l'industrie informatique, qu'il s'agisse de la valorisation des nombreuses inventions effectuées dans des centres de recherche, ou de la création de nouvelles entreprises par des chercheurs.

De ce point de vue, l'originalité de l'économie du logiciel réside dans l'invention de licences spécifiques, dont la plus connue est la *Licence Publique Générale* (GPL). En effet, pour une activité où, comme nous l'avons déjà analysée la distance peut être faible entre une invention et un produit commercialisable, ces licences en rendant obligatoire la diffusion du code-source, visent à interdire la transformation des logiciels libres en des progiciels commerciaux et donc empêchent le développement d'une trajectoire du monde de la création vers le monde fordiste¹³. Par contre, elles permettent le déploiement d'activités marchandes basées sur des logiciels libres, ce dont témoignent les succès - y compris financiers - des sociétés qui se sont créées autour de Linux. L'existence des logiciels libres a favorisé l'extension du monde de la production flexible, dans sa composante constituée de prestations diverses intégrant des services sur mesure à des logiciels. La fourniture de composants logiciels libres pourrait renforcer cette trajectoire en favorisant le développement de logiciels sur mesure à partir de composants logiciels standardisés (deuxième modalité du monde de production flexible), dont les potentialités semblent particulièrement importantes.

Section II - Des évolutions dynamiques différentes selon les zones géographiques de production des logiciels

Ces évolutions dynamiques, et notamment l'intensité variable de l'industrialisation de l'économie du logiciel, expliquent l'importance relative différente des mondes de production du logiciel entre les Etats-Unis d'une part, l'Europe occidentale et le Japon d'autre part. Nous nous limiterons à l'étude de ces trois zones qui représentent encore plus de 90 % du marché

¹² Les exemples de ce type d'évolution sont les plus spectaculaires aux Etats-Unis (par exemple la création de Sun par des chercheurs de Berkeley) mais il en existe également en France, comme les petites entreprises de génie logiciel issues des recherches menées par l'INRIA (Eurostaf, 1996, p.126).

¹³ C'est ce qui s'était produit pour certains logiciels publics, non couverts par des licences, et qui sont devenus des progiciels "fermés" et commercialisés, comme Mosaïc, premier logiciel de navigation sur Internet.

mondial du logiciel et des services informatiques, même si leur importance relative diminue sensiblement (de 91,4 % en 1994 à 90,3 % en 1996).

Tableau LIV
Répartition du marché mondial du logiciel et des services informatiques

	1994		1996	
	Milliards de FF	en %	Milliards de FF	en %
Etats-Unis	647	43,8	790	45,4
Europe Occidentale	484	32,7	540	31,0
Japon	221	14,9	241	13,9
Europe de l'Est	9	0,6	12	0,7
Reste du monde	118	8,0	157	9,0
Total monde	1 479	100,0	1 740	100,0

Source : EITO, in Eurostaf, 1996 A, p. 20.

A - LA DOMINATION DES ETATS-UNIS GRACE A UNE INDUSTRIALISATION PLUS RAPIDE DE L'ECONOMIE DU LOGICIEL

La force des Etats-Unis dans l'économie du logiciel s'explique principalement par leur domination dans le domaine des progiciels. En 1993, aux Etats-Unis les progiciels représentaient 29 % des dépenses consacrées au logiciel par les utilisateurs des technologies de l'information, alors qu'ils ne représentaient que 10 % au Japon et 24 % en Europe (OCDE, 1998 A, p. 7)¹⁴. Mais c'est surtout en termes d'offre de progiciels que la suprématie des entreprises américaines est écrasante, ce que montre l'analyse des échanges internationaux de progiciels. Certes la mesure des échanges internationaux de logiciels est délicate : selon les cas, ils peuvent être enregistrés comme des exportations de biens (quand le progiciel est

¹⁴ David C. Mowery à partir de données OCDE et IDC de 1985, estime que les progiciels représentaient 70 % du marché aux Etats-Unis, 50 % en Europe et 10 % au Japon et que si la croissance des progiciels était au départ plus forte aux Etats-Unis, elle était devenue plus rapide en Europe et au Japon (1996, p. 5 et 7).

vendu avec un ordinateur exporté), comme des exportations de services professionnels (quand il est vendu à une firme étrangère qui le commercialise) ou comme des revenus d'investissement (quand il est vendu par une branche de la firme située à l'étranger) et les statistiques des Douanes basées sur les supports qui passent la frontière n'appréhendent que partiellement la réalité des échanges effectués¹⁵. S'il faut donc interpréter les données avec prudence, "plus comme un ordre de grandeur que comme des données précises" (OCDE, 1998 A, p. 27), les différences constatées entre les Etats-Unis d'une part, l'Europe et le Japon d'autre part, sont suffisamment importantes pour être éloquents.

Tableau LV
Décomposition du marché mondial des progiciels en 1991
(en millions de dollars)

Origine des produits consommés	Zones de consommation				Total des produits consommés par région d'origine	Part de la région d'origine dans la consommation mondiale
	Etats-Unis	Europe	Japon	Reste du monde		
Etats-Unis	19,93	13,28	2,95	3,81	39,98	78,3 %
Europe	0,49	7,38	0,37	0,00	8,24	16,1 %
Japon	0,00	0,12	2,09	0,00	2,21	4,3 %
Reste du monde	0,12	0,37	0,00	0,12	0,62	1,2 %
Consommation de la région	20,54	21,16	5,41	3,94	51,05	100,0 %
Part de la consommation mondiale	40,2 %	41,4 %	10,6 %	7,7 %	100,0 %	

Source : Données IDC, in OCDE, 1998 A, p. 30

¹⁵ Par exemple aux Etats-Unis, les données sur les échanges sont collectées à partir du contenu de la déclaration de l'expéditeur à l'exportation, mais qui n'est obligatoire que pour des produits de plus de 2 500 dollars. De ce fait, l'exportation de nombreuses copies d'un progiciel dont la valeur unitaire est inférieure à cette somme peut ne pas être appréhendée. De même, les compagnies américaines peuvent exporter une copie unique d'un disque (la "copie d'or") directement à un distributeur dans un pays étranger, qui réalise alors des copies autorisées du logiciel et les vend aux utilisateurs finaux du produit, transactions qui n'apparaîtront pas dans les données sur les échanges (OCDE, 1998 A, p. 46). Ajoutons que le développement récent de la vente de logiciels téléchargés sur Internet va rendre les échanges internationaux encore plus difficiles à mesurer.

En effet, le contraste est saisissant entre les Etats-Unis où les produits d'origine intérieure couvraient 97 % du marché intérieur, et l'Europe et le Japon où les taux de couverture étaient respectivement de 34,9 % et de 38,6 %. De plus, les exportations de logiciels des Etats-Unis ont fortement augmenté au cours des années 90, avec un taux de croissance moyen de 23 % par an entre 1990 et 1994 (OCDE, 1997 B, p. 61).

Parmi les sociétés européennes produisant des logiciels, les plus importantes sont des sociétés françaises et allemandes. Le secteur du logiciel français a longtemps été "un sujet de fierté nationale et une source traditionnelle d'emplois à forte valeur ajoutée" (Club de l'Arche, 1993, p. 5). Dans le domaine des logiciels sur mesure et des services informatiques, certaines SSII françaises fortement concentrées et ayant pour principaux actionnaires des banques ou des grandes entreprises et des organismes publics, se sont rapidement développées et ont réussi à s'implanter sur les marchés internationaux (Pascal Petit, 1994, p. 301)¹⁶. Mais si les entreprises françaises disposent d'un parc logiciel encore largement dominé par les logiciels spécifiques, la part des progiciels augmente rapidement : elle passe de 32 % de la dépense informatique française en 1994 à 42,5 % en 1999 (Eurostaf, 1996 A, p. 94). Or à l'exception notable du segment particulier constitué par les progiciels de jeux, et d'une entreprise, Business Objects, pour les progiciels d'aide à la décision, peu d'entreprises françaises ont réussi à s'imposer sur les marchés des progiciels. Les principaux éditeurs de progiciels en France sont américains (IBM, Microsoft, Computer Associates, Oracle) et si, aux logiciels importés des Etats-Unis, on ajoute ceux développés en France par des entreprises américaines, le total atteint 70 % de la valeur du marché français des progiciels (OCDE, 1998, p. 28). Selon le Club de l'Arche, "le secteur du logiciel français traverse aujourd'hui une crise profonde, (...) essentiellement structurelle, avec une offre centrée sur les services et le "sur mesure", alors que la demande s'oriente inexorablement vers le "prêt à l'emploi" et les produits standardisés" (1993, p.5)¹⁷.

¹⁶ Robert Salais et Michael Storper estiment que la plus importante d'entre elles, Cap Gemini Sogeti, est le seul exemple en France, "d'une PME qui ait réussi à devenir une grande firme de haute technologie" (1993, p. 229).

¹⁷ De façon quelque peu catastrophiste, le Club de l'Arche souligne qu'il est indispensable "d'aider ce secteur à sortir de l'impasse, impasse qui, si l'on n'y prend garde, risque de se transformer dans ce secteur en une crise analogue à celle de la sidérurgie, avec, à la clef, des dizaines de milliers de chômeurs supplémentaires" (1993, p. 5).

En Allemagne, les SSII se sont développées dans les années soixante et soixante dix avec des liens particulièrement étroits avec les entreprises industrielles. Les sociétés allemandes sont moins concentrées et moins internationalisées que les sociétés françaises : la plupart sont fortement tournées vers le marché intérieur et dépendent d'un petit nombre de clients et les quelques deux mille sociétés de services informatiques allemandes sont en concurrence avec plus de dix mille petites sociétés indépendantes qui sont souvent des "artisans" de l'informatique (Eurostaf, 1996 A, p. 40-41). Par contre une entreprise allemande (SAP) a réussi à devenir un des principaux producteurs de progiciels de gestion d'entreprise.

Une idée très répandue est celle de la faiblesse intrinsèque du Japon en matière de logiciels, comparée à sa réussite en électronique. Par exemple, Philippe Breton estime que "l'une des caractéristiques de l'informatique japonaise, très axée sur l'électronique, a été longtemps sa pauvreté en matière de production de logiciels, pratiquement tous importés jusqu'à une date récente" et va jusqu'à se demander s'il n'existe pas un "facteur culturel", "les programmeurs japonais n'auraient, dit-on, guère de sympathie pour la stricte logique binaire" (1987, p. 199). En réalité, les insuffisances japonaises concernent uniquement les progiciels¹⁸, le "développement du logiciel [étant] contrôlé par les constructeurs qui privilégient les programmes sur mesure pour les ordinateurs centraux plutôt que des solutions standardisées pour les micro-ordinateurs et les réseaux de stations de travail" (Charles Goldfinger, 1994, p. 53). De plus, les logiciels japonais sont beaucoup plus produits "en interne" que dans les autres pays, ce que confirment les résultats d'une étude sur les programmes d'application utilisés par les entreprises japonaises : "deux millions de programmes sont des programmes "maison", plus d'un demi-million sont fournis par des sociétés de logiciels et moins de 250 000 sont créés par des constructeurs d'ordinateurs" (OCDE, 1993, p. 94). En conséquence, les progiciels ne représentent que 15 % de l'ensemble du marché des logiciels japonais en 1995, proportion qui a peu varié depuis 1990 (OCDE, 1997 B, p. 22), et la récente accélération de la croissance du marché des progiciels japonais due à la demande soutenue d'ordinateurs personnels et au succès croissant du multimédia, a surtout profité aux entreprises américaines (idem).

¹⁸ Jean-Pierre Brulé estime que "les Japonais n'ont toujours pas conçu un seul progiciel d'envergure mondiale" (1993, p.292).

B - EXPLICATIONS ET PERSPECTIVES

Comment expliquer que l'industrialisation de l'économie du logiciel ait été plus précoce et plus rapide aux Etats-Unis qu'en Europe ou au Japon ?

Une explication souvent avancée est que le régime de protection des droits de la propriété en matière de logiciels, est plus fort aux Etats-Unis qu'en Europe et au Japon. Logiquement, les tenants de cette explication proposent de renforcer cette protection, par exemple en donnant la possibilité de déposer des brevets sur des logiciels, pour stimuler la production des progiciels. Au contraire, Robert Merges montre, dans une étude comparative sur les droits de propriété dans l'économie du logiciel, que c'est plus la structure de l'industrie qui influence le régime légal que l'inverse (1996, p. 282). Il insiste sur le nombre de facteurs en interdépendance, et sur l'impossibilité d'isoler un seul facteur explicatif, ce qui rejoint les approches en termes de systèmes nationaux d'innovations (Jorge Niosi, Bertrand Bellon, Paolo Saviotti, Michaël Crow, 1992). Ces approches expliquent la diversité des trajectoires nationales à partir de l'existence de schémas d'interrelations ayant une certaine cohérence, qui se sont constitués historiquement différemment, en soulignant que "ce n'est pas un type d'institutions qui crée un système, mais un ensemble d'interactions entre la série d'institutions, ainsi que les réponses qu'elles reçoivent de leur environnement" (idem, p. 235-236).

Ainsi au Japon, le succès des entreprises japonaises dans les différents segments du matériel informatique s'est opéré avec des architectures concurrentes. Cette "fragmentation architecturale", caractérisée par l'absence d'un modèle dominant dans le matériel, a découragé le développement de progiciels (David C. Mowery, 1996, p. 308). Les logiciels sur mesure sont développés au sein de l'entreprise ou par des entreprises indépendantes faisant partie du groupe (*keiretsu*), chaque entreprise de logiciel ayant tendance à fournir une seule entreprise (Robert Merges, 1996, p. 278). Ce modèle est cohérent avec une protection légale faible, qui facilite le transfert de technologies de pays étrangers. Ce n'est qu'en 1985 et sous la pression des Etats-Unis qu'est effectuée une intégration du logiciel à la loi sur le copyright, et cette protection connaît des limites significatives. Par contre, les logiciels sur mesure étant développés dans de grandes structures très performantes ("usines à logiciel"), la différence entre les prix des logiciels sur mesure et les prix des progiciels est moins importante qu'ailleurs (OCDE, 1994, p. 31). En termes de trajectoires, on peut dire que c'est le succès momentané des entreprises japonaises sur une forme d'industrialisation "traditionnelle" qui a

constitué un obstacle pour accéder à la forme d'industrialisation la plus prometteuse, à savoir la substitution de progiciels à des logiciels sur mesure.

A l'inverse la taille du marché américain (en termes de nombres d'ordinateurs installés d'un même type) et les "avantages dus à la flexibilité des universités américaines" ont accentué pour les entreprises américaines de progiciels les avantages dont disposent les *first movers* (David C. Mowery, 1996, p. 305). L'Europe se caractérise par une situation intermédiaire entre les Etats-Unis et le Japon. En effet, s'il existait, comme au Japon, des constructeurs nationaux incompatibles soutenus par leur gouvernement ce qui pouvait constituer des niches pour des producteurs de logiciels, ce facteur a été moins important vu la faiblesse de l'industrie du matériel informatique européenne, et le marché européen du logiciel est rapidement devenu beaucoup plus ouvert (David C. Mowery, 1996, p. 310). La question qui se pose est de savoir s'il faut continuer à essayer de transformer - comme cela a été fait en vain jusqu'à maintenant - des sociétés de services informatiques européennes en éditeurs de progiciels qui devront affronter la concurrence d'entreprises principalement américaines solidement installées, ou s'il ne serait pas préférable de les encourager à développer des activités qui permettent d'apporter des solutions personnalisées autour des progiciels existants (trajectoire "d'objectivation") ? Et s'il peut exister des craintes quant aux dépendances excessives que créerait le fait que les progiciels sont développés par des entreprises étrangères, le meilleur moyen de les surmonter n'est-il pas de créer ces prestations autour de logiciels libres, ce qui conduit à s'interroger sur la poursuite de leur développement ?

Section III - Une illustration : potentialités et difficultés du développement du monde de production des logiciels libres

Nous avons vu que les logiciels libres se sont développés quasi spontanément dans plusieurs domaines où ils connaissent un certain succès. Si on peut penser qu'une extension éventuelle des logiciels libres serait bénéfique pour l'ensemble de l'économie du logiciel (B), elle ne s'effectuera peut-être pas aussi facilement (C). Auparavant, il est nécessaire d'effectuer un premier bilan des conséquences du développement actuel des logiciels libres sur l'ensemble de l'économie du logiciel (A).

A - L'IMPACT DES LOGICIELS LIBRES : CONCURRENCE SUR CERTAINS SEGMENTS DES PROLOGICIELS ET DEVELOPPEMENT DES SERVICES LIES

Nous avons vu que la production de progiciels en grande série, par la répartition des coûts de conception sur l'ensemble des exemplaires produits, s'est révélée très efficace dans la couverture de besoins standard mais que la production commerciale des progiciels, en raison de l'importance des rendements croissants d'adoption, entraîne à moyen terme une concentration monopolistique. La présence de nombreux producteurs de progiciels s'expliquait surtout par l'existence de sous-secteurs relativement indépendants correspondant à des progiciels aux fonctionnalités très différentes, s'exécutant sur des matériels divers et qui n'étaient pas en concurrence les uns avec les autres (absence de substitution possible entre un traitement de texte et un tableur ou entre un progiciel de gestion s'exécutant sur micro-ordinateur ou sur mini-ordinateur) ; de plus l'apparition périodique de nouveaux domaines d'application entraîne une vive concurrence entre de multiples producteurs, parfois de taille très réduite, même si à terme ne subsistait principalement qu'un producteur leader. Cependant, le rapprochement entre les différents types de matériels, le contrôle légal ou technique des standards notamment pour les interfaces fonctionnelles des logiciels et la représentation des informations, et la puissance commerciale et financière des principaux éditeurs de progiciels (notamment Microsoft) font que cette concentration a tendance à s'étendre à l'ensemble du secteur des progiciels voire à dépasser ce secteur, avec le rapprochement de technologies auparavant relativement séparées (l'informatique, les télécommunications, et plus récemment la télévision) et le développement de nouvelles externalités de réseaux directes et indirectes qui en résulte. *Les risques d'une insuffisante diversité des progiciels sont à terme importants* : "l'écologie des idées et des techniques obéit aux mêmes lois que celles des êtres vivants, la quasi-unicité des solutions présente plusieurs dangers" (Bernard Lang, 1998 A). Elle peut diminuer la quantité et la variété des recherches et donc le progrès technique : "une fois la concurrence disparue, le seul producteur restant n'a plus aucun intérêt à investir pour améliorer ses produits" (Bernard Lang, 1997, p. 2). Plus précisément, le développement et la commercialisation de nouveaux produits ou de nouvelles versions obéissent uniquement aux intérêts du producteur et peuvent rendre les utilisateurs dépendants du producteur, notamment par l'imposition de fait de standards propriétaires (exemple des formats de fichiers de Microsoft). A l'opposé, les caractéristiques de la création et de l'amélioration permanente des

logiciels libres, exposées précédemment, ont permis la production de logiciels diversifiés¹⁹ et de qualité dans certains secteurs des logiciels et ont contribué au maintien ou au rétablissement d'une situation de concurrence sur ces segments (systèmes d'exploitation, serveurs de bases de données...) ce qui se traduit notamment dans l'évolution des prix. Par exemple, dans le domaine des systèmes d'exploitation pour serveurs, la dynamique qui s'est créée autour de Linux fait qu'il constitue maintenant une alternative crédible à Microsoft, même si son premier effet a été d'abord "d'accélérer la disparition des architectures et des systèmes d'exploitation propriétaires et des Unix non libres, hors Microsoft" (Christian Genthon, Denis Phan, 1999, p. 187).

La deuxième conséquence du développement des logiciels libres a été d'étendre la fourniture de prestations personnalisées accompagnant des logiciels, en permettant de proposer des solutions plus variées et moins coûteuses quand elles sont basées sur des logiciels libres (cf. chapitre VII).

B - DES PERSPECTIVES INTERESSANTES...

Ce bilan provisoire peut être jugé satisfaisant. Les effets positifs des logiciels libres peuvent se prolonger dans deux directions : l'extension à l'ensemble du champ des logiciels et le développement de composants logiciels libres. La première direction n'appelle pas de commentaires particuliers. On peut simplement noter que la crédibilité d'un logiciel est très dépendante de l'existence de logiciels complémentaires (cf. le succès différencié de Linux en tant que serveur d'entreprise et en tant que système d'exploitation d'un micro-ordinateur utilisé principalement pour de la bureautique). Certes, il peut exister des solutions comportant des logiciels libres et des logiciels commerciaux, mais les qualités spécifiques des logiciels libres ne sont pas limitées a priori à certains domaines d'application.

¹⁹ On retrouve la même argumentation de maintien de la diversité grâce à un contexte non-marchand, appliqué au cadre plus général de la recherche scientifique dans l'analyse de Michel Callon (1994 B). De même, Giovanni Dosi et Luigi Orsenigo expliquent que "une sélection très efficace peut parfois entraver le démarrage et le processus de diffusion des nouvelles technologies lorsque ces dernières ne paraissent pas vraiment rentables à l'origine. Les expériences sans succès sont punies sévèrement et rapidement, de sorte qu'il peut arriver que certaines technologies ouvrant des perspectives d'avenir intéressantes ne disposent pas de suffisamment de temps pour se développer et être perfectionnées" (1988, p. 30). Enfin Abbe Moshowitz estime que "la variété est la première victime de la marchandisation des produits informationnels" et que "l'actuelle profusion sera réduite, par les exigences de la concurrence sur le marché, à un petit nombre d'offres dans chaque catégorie" (1997, p. 34).

La deuxième direction correspond à un des enjeux majeurs de l'économie du logiciel. Face à l'extension permanente de l'utilisation de logiciels et de la diversité des besoins, le développement de logiciels sur mesure à partir de composants standardisés (deuxième modalité du monde de production flexible) apparaît comme une solution permettant de concilier une productivité et une fiabilité relativement élevée (par la réutilisation de composants testés) tout en pouvant ajuster le logiciel développé aux besoins précis des utilisateurs. Mais le développement de cette forme de production particulièrement prometteuse ne va pas de soi. Pour être pleinement efficace, il faudrait pouvoir utiliser des composants logiciels choisis parmi l'ensemble de ceux qui ont été développés par la communauté informatique, et ne pas se limiter à ceux qui ont été produits en interne. Le problème est que l'utilisation des composants logiciels nécessite l'accès direct au code source de ces composants pour résoudre les problèmes d'interopérabilité, de fiabilité et de pérennité. Mais permettre de disposer librement du code source des composants empêche le paiement de l'accès à ces composants. Ce dilemme, qui n'est pas sans rappeler le dilemme de la connaissance de Fritz Machlup (1984)²⁰, explique pourquoi des marchés de composants logiciels ont tant de difficultés à émerger. L'existence de ces composants sous forme de composants logiciels libres pourrait constituer une solution intéressante, mais rien ne garantit que ce modèle se développera spontanément.

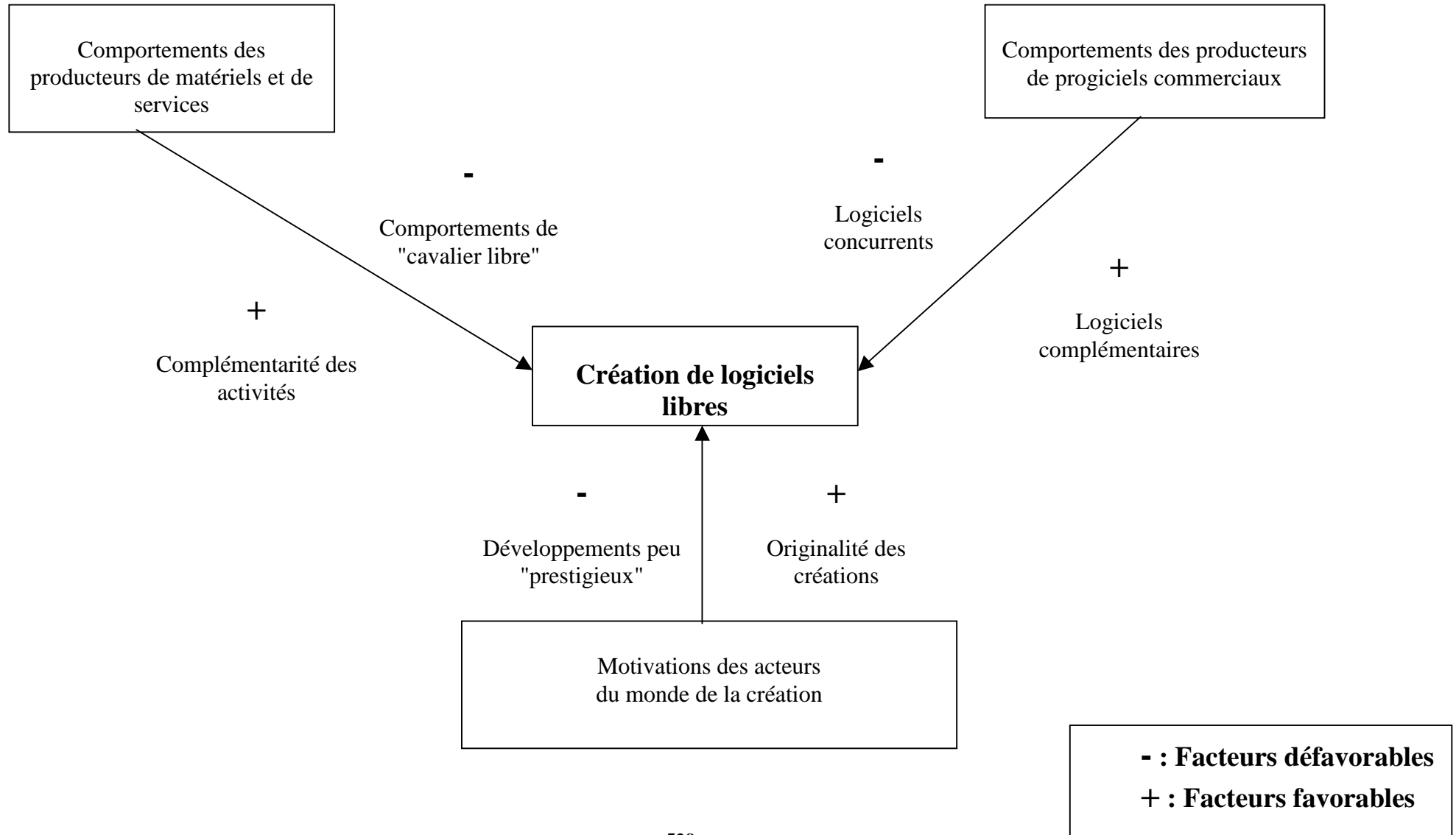
C - MAIS DIFFICILEMENT ACCESSIBLES SPONTANEMENT

Si les perspectives offertes par un nouveau développement et une extension du champ des logiciels libres apparaissent très prometteuses, il n'est nullement évident que cette évolution se réalise spontanément²¹. En effet, de façon apparemment paradoxale, l'essor actuel des logiciels libres risque d'aboutir à une situation où les facteurs favorables sont dominés par des facteurs défavorables à leur développement (cf. schéma XIII).

²⁰ Fritz Machlup formule ainsi ce dilemme : "seul l'espoir d'un prix positif pour l'utilisation du savoir peut garantir l'allocation des ressources vers la création du savoir, mais seul un prix nul peut garantir une utilisation efficiente du savoir une fois qu'il a été créé". Dans le cas des composants logiciels, on pourrait dire que "seule la possibilité de facturer un composant logiciel, grâce à la *fermeture* du code source, peut garantir l'allocation des ressources vers la création de composants logiciels, mais que seule *l'ouverture* du code source peut garantir une utilisation efficiente des composants logiciels une fois qu'ils ont été créés". Les analogies entre des problèmes d'économie de la connaissance et d'économie du logiciel sont du reste peu surprenantes, dans la mesure où un logiciel est constitué de connaissances qui ont été codifiées.

²¹ Rappelons que jusqu'à maintenant le développement des logiciels libres n'a pas été le résultat d'un projet délibéré des pouvoirs publics, même si leur rôle indirect a été loin d'être négligeable.

Schéma XIII : Les facteurs de la création de logiciels libres



Le facteur le plus important, qui explique la naissance et les premiers développements des logiciels libres, est l'existence de puissantes motivations non marchandes pour produire de tels logiciels, que nous avons analysées dans le chapitre précédent. Par contre, les dynamiques d'autorenforcement qui se sont créées autour de certains logiciels libres, au point de les faire apparaître comme des alternatives crédibles aux logiciels commerciaux dans certains domaines, s'expliquent également par le comportement d'autres acteurs informatiques, producteurs de logiciels commerciaux et producteurs de matériels et de services complémentaires.

Tout d'abord des sociétés de services se sont créées pour commercialiser des services complémentaires liés à des logiciels libres. Ces sociétés, qui le plus souvent participent à l'évolution des logiciels sur lesquels sont basées leurs prestations, connaissent une croissance parallèle à celle des logiciels libres, croissance qu'elles ont favorisée en élargissant le cercle des usagers potentiels au-delà des utilisateurs experts initiaux, ou qui disposaient "en interne" des compétences nécessaires.

Mais "si le libre se développe aujourd'hui, c'est aussi parce qu'il s'insère dans les stratégies des grands acteurs du monde informatique" (Christian Genthon, Denis Phan, 1999, p. 182). Un certain nombre d'initiatives ou d'annonces constituent un soutien important à Linux : vente par des constructeurs d'ordinateurs (Dell, HP, Compaq) de micro-ordinateurs et de serveurs avec Linux préinstallé, prises de participation financières dans les sociétés de service qui se sont constituées autour de Linux (par exemple Intel et Netscape dans Red Hat), développement de versions de logiciels applicatifs pour Linux (le traitement de texte WordPerfect de Corel, la suite bureautique StarOffice de Star Division racheté récemment par Sun, les principaux systèmes de gestion de bases de données pour grands systèmes²², le logiciel de création de page Web Golive), développement de "pilotes" pour périphériques par certains producteurs comme Adaptec un des principaux fournisseurs de cartes SCSI²³. Une illustration de l'ampleur de ce soutien est la création de Linux International, présidé par John

²² L'annonce par le leader Oracle de la disponibilité de son SGBDR sous Linux a été suivie en quelques mois par les autres principaux éditeurs (Informix, Sybase, Computer Associates) qui proposent désormais tous des versions sous Linux téléchargeables gratuitement (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 68).

²³ Cette question est importante. Une des faiblesses de Linux était la difficulté pour trouver les "pilotes" logiciels nécessaires pour utiliser des périphériques, les producteurs de ceux-ci ne les fournissant pas, voire même refusant de livrer les spécifications indispensables pour leur développement par des tiers.

Hall, senior manager chez Digital-Compaq, dont les objectifs sont de diffuser les informations concernant Linux, d'assurer sa promotion et de faire tester ces dernières versions. Cette organisation regroupe les principales sociétés qui se sont constituées autour de Linux (SuSe, Caldera, Red Hat...) et des acteurs "traditionnels" de l'informatique (SCO, Compaq, Silicon Graphics, Sun, Informix...). IBM a également apporté son soutien à certains logiciels libres en annonçant des versions sous Linux de ces logiciels (son gestionnaire de bases de données DB2, son système de fichiers en réseau AFS, et les progiciels développés par sa filiale Lotus), et l'intégration d'un logiciel libre, Apache, logiciel de serveur Web le plus diffusé, à son offre d'*e-business* WebSphere²⁴.

Cependant, l'ampleur de ces soutiens à certains logiciels libres (essentiellement Linux et dans une moindre mesure Apache) ne doit pas masquer l'importance des facteurs qui peuvent jouer défavorablement sur l'extension des logiciels libres.

Premièrement, la motivation pour les acteurs du "monde de la création" à développer des logiciels libres risque d'être beaucoup moins importante pour des activités moins exaltantes et moins prestigieuses que les créations déjà effectuées²⁵, et pour la création de produits dont les producteurs ne sont pas les utilisateurs principaux initiaux (logiciels de bureautique et de gestion, bibliothèques publiques de composants logiciels pour les applications métiers...). De même, si de manière informelle, des organisations publiques, mais aussi dans certains cas privées, laissent des informaticiens contribuer au développement de logiciels libres sur leur temps de travail, il n'est pas évident que ces organisations maintiennent une attitude aussi bienveillante si elles ne bénéficient pas directement des logiciels ainsi développés.

Deuxièmement, la croissance des logiciels libres élargit le marché des producteurs de matériel et de services complémentaires. Mais cet élargissement peut entraîner l'augmentation des comportements de "cavalier libre", à savoir profiter de l'existence de logiciels libres sans

²⁴ IBM éprouva quelques difficultés pour négocier avec les développeurs d'Apache, qui sont des programmeurs répartis aux quatre coins du globe. "Même si on voulait payer, il n'y avait personne à payer, les auteurs n'existent pas, légalement" constate un des dirigeants d'IBM (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 66-67).

²⁵ Le prestige d'une activité est notamment lié à son caractère novateur : développer un éditeur de texte programmable dans les années 70 (comme "Emacs" réalisé au MIT) pouvait contribuer à la renommée de ses auteurs ; ce n'est certainement plus le cas actuellement.

participer à leur développement, en raison d'une série de phénomènes liés : apparition d'acteurs à la culture très éloignée du logiciel libre à la différence des sociétés de services initialement présentes, transformation possible des valeurs culturelles de ces sociétés en conséquence de leurs succès²⁶, durcissement de la concurrence sur ce marché et vente à des utilisateurs plus diversifiés, moins sensibilisés aux enjeux des logiciels libres et pour lesquels le prix de la solution proposée est un critère plus important que l'image de marque de la société, compétitivité supérieure des entreprises qui ne supportent pas de coûts de développement des logiciels libres.

Ainsi Jean-Paul Smets-Solanes et Benoît Faucon soulignent que "l'intégration croissante de l'industrie et des organisations de promotion des logiciels libres comporte des risques de dérive non négligeables : le financement des organisations et des projets structurés autour des logiciels libres (...) par des industriels pourrait à terme infléchir les développements de ces produits en fonction des besoins des industriels et non de ceux des utilisateurs" (1999, p. 134). Pour les sociétés "éditrices" ou "distributrices" de logiciels libres comme Red Hat ou SuSe, à l'heure actuelle, les innovations qu'elles introduisent en incorporant des fonctionnalités qui leur sont propres profitent également à leurs concurrents²⁷. Le risque est qu'il se produise une divergence entre les distributions, en ne tenant pas compte des fonctionnalités introduites par des concurrents, aboutissant à une fragmentation du marché entre plusieurs solutions imparfaitement compatibles, comme cela s'est produit pour Unix (cf. chapitre II)²⁸.

Troisièmement, l'extension du champ des logiciels libres à de nouveaux produits risque d'atténuer voire d'annuler le soutien que pouvait apporter certains éditeurs de progiciels

²⁶ Par exemple, la Red Hat Software, principal "distributeur" de Linux, a été introduite avec succès à la bourse de New York le 11 août 1999 : le cours de l'action a été multiplié par six, trois jours après son introduction.

²⁷ "Grâce à l'ouverture du code source, SuSe peut profiter des innovations que j'apporte à ma distribution" déclare Bob Young, président de la Red Hat (entretien avec Jean-Paul Smets-Solanes et Benoît Faucon, 1999, p. 97).

²⁸ Par exemple, il existe une initiative pour standardiser le contenu des distributions, *Linux Standard Base* (LSB) mais un acteur comme Red Hat refuse (pour le moment ?) d'y participer (cf. "Linux backers seek to avoid Unix-like splintering", <<http://www.infoworld.com/cgi-bin/displayStory.pl?99035.whlinser.htm>>).

(souvent en position de faiblesse)²⁹ si les logiciels libres, de produits complémentaires à leurs propres produits, deviennent des produits concurrents.

Quatrièmement, la menace que peuvent constituer les logiciels libres pour les intérêts de certains producteurs de progiciels commerciaux puissants, suscite des réactions, notamment en utilisant "l'arme des brevets" (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 141). Le durcissement de la protection des logiciels par le recours aux brevets peut s'expliquer par le fait qu'avec le développement d'Internet, les stratégies de protection par renouvellement accéléré des produits deviennent moins efficaces³⁰. A propos des nouveaux problèmes de gestion de la propriété intellectuelle que poserait l'existence d'Internet, Carl Shapiro et Hal R. Varian font toutefois remarquer que l'apparition de chaque nouvelle technologie de reproduction, que ce soit la presse à imprimer ou le magnéto-scope s'accompagne de prévisions catastrophiques qui ne se sont jamais réalisées (1999, p. 80)³¹. En réalité le dépôt de brevets peut être utilisé pour empêcher le développement des logiciels libres, ce que reconnaissait explicitement Microsoft dans un rapport interne, surnommé *Mémoire d'Halloween* (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 72). S'il n'existe pas encore d'exemple d'utilisation des brevets contre des logiciels libres, "il est possible que d'ici à cinq ans, il ne soit plus possible de se servir d'un système comme Linux, à cause des brevets" estime Richard Stallman (fondateur de la *Free Software Foundation*). De façon plus pernicieuse, les brevets peuvent permettre de créer des logiciels qui n'ont guère de libres que le nom, en gardant le contrôle sur des logiciels, tout en publiant leur code source. C'est notamment la stratégie de Sun avec le langage Java ou avec la décision d'ouvrir les sources du logiciel de navigation Communicator 5 sous une licence publique spécifique (la *Netscape Public Licence*). Cette tentative n'a toutefois guère suscité d'intérêt de la part des développeurs extérieurs à Netscape dans la mesure où le programme devait rester la propriété de Netscape (Jean-Michel Dalle, Nicolas Jullien, 2000, p. 7). Par contre, certains défenseurs du logiciel

²⁹ Sans faire de faux procès, on peut penser que la volonté de contrecarrer la volonté hégémonique de Microsoft a joué un rôle important dans certaines prises de position qui pourraient se modifier en cas d'affaiblissement de Microsoft.

³⁰ Ainsi Anne Mayère estime que "la meilleure protection de l'information passe par son renouvellement" (1997, p. 131). L'obsolescence accélérée des progiciels peut être mise en parallèle avec les pratiques des consultants qui renouvellent rapidement leurs méthodes pour se protéger.

³¹ Ainsi, Carl Shapiro et Hal R. Varian indiquent qu'au Moyen Age, les professeurs avaient coutume de faire leurs cours dans des pièces sombres, pour éviter que les étudiants puissent prendre des notes (idem, p. 88).

libre, avec à leur tête Eric S. Raymond, ont créé l' *Open Source Initiative* dont l'objectif est de défendre le label *Open Source* contre d'éventuelles récupérations, par une certification "des projets et des logiciels répondant effectivement au modèle de développement à code source ouvert" (Jean-Paul Smets-Solanes, Benoît Faucon, 1999, p. 127).

L'essor actuel des logiciels libres risque donc d'aboutir à une situation où les facteurs favorables sont dominés par les facteurs défavorables à leur création. En conséquence, un soutien délibéré des pouvoirs publics pourrait s'avérer indispensable à un nouveau développement du logiciel libre. Ce soutien peut consister en différentes actions : à l'inverse des tentatives d'étendre le champ des brevets aux logiciels, créer "une sorte de répertoires de bonnes idées que personne ne pourrait s'approprier" (Philippe Riviere, 1999, p. 23)³² ; inciter à l'utilisation des logiciels libres dans les administrations (notamment dans l'enseignement), ce qui fait l'objet actuellement de deux projets de lois en France à l'Assemblée Nationale et au Sénat ; a minima imposer pour les informations fournies ou demandées par les administrations, l'utilisation de formats publics et non privés comme c'est fréquemment le cas actuellement ; octroyer des subventions à des équipes développant des logiciels libres, selon un modèle proche de celui en vigueur dans la recherche scientifique³³ ; aider au développement et au fonctionnement d'institutions (du type de celles à la base d'Internet) pour maintenir la cohérence des développements, la certification des logiciels et des composants. Une question particulièrement importante est celle de l'initialisation d'un projet de développement de logiciel (ou de composant logiciel) libre et notamment de la définition de ces spécifications. En effet, si le modèle de développement très décentralisé des logiciels libres s'est révélé particulièrement efficace pour l'écriture et le débogage des logiciels, il

³² En constatant que le plus souvent face à une plainte pour violation de brevet, la riposte la plus efficace consiste à contre-attaquer en déposant en retour une plainte pour violation d'un brevet que l'on possède, Jean-Paul Smets-Solanes et Benoît Faucon préconisent la création de "fonds de brevets libres défensifs" où seraient mis en commun les brevets déposés par les instituts de recherche public, les universités, et les partisans des logiciels libres. Ces brevets comprendraient une clause contractuelle spécifique précisant que n'importe quel auteur de logiciel libre peut obtenir automatiquement et gracieusement une licence d'exploitation de ces brevets (1999, p. 151-152).

³³ Subventionner la production de logiciels libres peut être justifié par certaines conclusions de Kenneth J. Arrow (1962) sur la "production de connaissances" où il proposait de séparer la rémunération de l'innovateur des charges qui pèsent sur les consommateurs bénéficiaires de l'innovation. En effet, dans le cas des connaissances comme des compétences, la mise en œuvre de ces ressources a pour effet immédiat d'en accroître la disponibilité et non de la diminuer. Une subvention de ces inputs, de façon à faire baisser leur prix, entraîne une utilisation plus intensive qui en augmente la disponibilité, alors qu'habituellement, seule la hausse du prix d'un input rare entraîne cette augmentation (cf. également Albert O. Hirschman, 1986, p. 103). Il nous semble que ce raisonnement s'applique particulièrement bien aux logiciels.

présente plus de difficultés pour initier les spécifications (Christian Genthon, Denis Phan, 1999, p. 180), ce qui explique que ce sont des projets où les concepteurs étaient les premiers utilisateurs qui ont réussi à franchir cette étape. Si l'on souhaite que le champ des logiciels libres puisse s'étendre à des domaines comme les logiciels bureautiques et de gestion, ou les bibliothèques publiques de composants logiciels, il est certainement nécessaire de mener une action volontariste pour aider à la constitution d'organisations souples sur des projets permettant de définir des spécifications et d'harmoniser les développements effectués³⁴.

Les moyens nécessaires pour mener ces différentes actions, qui demandent un effort financier limité³⁵, doivent être rapportés à l'importance de l'enjeu pour l'évolution de l'ensemble de l'économie du logiciel.

La coexistence de mondes de production différents est un phénomène durable dans l'économie du logiciel, notamment en raison des très fortes complémentarités existant entre ces mondes. Ceci n'empêche pas que l'économie du logiciel connaît d'importantes évolutions dynamiques, mais ces changements se traduisent par l'existence de *plusieurs* trajectoires, correspondant au déplacement d'une production effectuée dans un monde de production vers un autre monde de production. A côté d'une trajectoire d'industrialisation, qui aboutit à une production en grande série de biens standard (monde "fordiste spécifique" des logiciels commerciaux), existent également des trajectoires de valorisation, de flexibilisation et d'objectivation, qui peuvent permettre un développement du monde de production flexible des logiciels, à partir des trois autres mondes de production.

³⁴ Eric S. Raymond (1998) précise que "au départ pour que le "modèle bazar" fonctionne, il faut être capable de présenter une promesse plausible". "L'écosystème des logiciels libres étant constitué de nombreux acteurs indépendants qui ont parfois du mal à se regrouper, l'action des pouvoirs publics devrait légitimement viser à mettre en réseau les compétences pour renforcer l'écosystème" (Jean-Paul Smets-Solanes et Benoît Faucon, 1999, p. 151).

³⁵ Dans certains cas (utilisation) ce soutien peut se traduire par des économies substantielles.

Cette analyse a permis de mettre en évidence la domination *actuelle* des Etats-Unis sur l'économie du logiciel, grâce à une industrialisation beaucoup plus rapide de leur production. Mais l'existence d'autres trajectoires, et notamment de celle que nous avons qualifiée de valorisation, peut contribuer à modifier cette situation. De ce point de vue, le développement du monde de production des logiciels libres, outre sa contribution au nécessaire maintien d'une certaine diversité des logiciels, peut être riche de potentialités pour favoriser l'avènement d'un véritable "sur mesure de masse" dans l'économie du logiciel, grâce à la production de composants logiciels libres. Toutefois, il n'est pas évident que l'extension du champ des logiciels libres se poursuivra aussi spontanément que cela a été le cas jusqu'à présent. L'analyse des différents facteurs du développement des logiciels libres montre que les facteurs défavorables à leur essor risquent de dominer les facteurs favorables. De ce point de vue, une action volontariste et multiforme des pouvoirs publics en faveur des logiciels libres pourrait être très bénéfique.

CONCLUSION

Dans cette thèse, nous avons montré :

1) que l'économie du logiciel possède deux particularités fondamentales, qui justifient une analyse spécifique, même si cette analyse ne peut être menée indépendamment de l'économie de l'informatique dont elle fait partie ;

2) que la première de ces particularités, à savoir la permanence des questions critiques auxquelles est confrontée l'économie du logiciel, ne peut être expliquée que *partiellement* par les spécificités techniques des logiciels et de leur processus de production ;

3) que la deuxième de ces particularités, à savoir la diversification croissante de l'économie du logiciel, ne peut être expliquée que *partiellement* par les problématiques habituelles de l'économie industrielle ;

4) que l'utilisation d'un cadre théorique socio-économique (la théorie des mondes de production) permet d'enrichir significativement l'analyse de l'économie du logiciel, notamment en permettant d'intégrer les deux dimensions précédemment citées. Il permet d'analyser *l'économie du logiciel comme étant fondamentalement marquée par la coexistence permanente de plusieurs mondes de production, qui reposent sur l'existence de conventions différentes, centrées sur un type de produit, et constituant des réponses différenciées aux questions critiques de l'économie du logiciel.*

5) que cette analyse peut permettre d'appréhender les dynamiques à l'œuvre dans l'économie du logiciel, et éventuellement d'agir en tentant d'influencer les évolutions anticipées.

La démonstration de l'idée centrale de cette thèse – la pluralité durable des mondes de production des logiciels – nous a conduit à mettre en évidence l'importance croissante des logiciels dans l'économie d'une part, les spécificités des logiciels d'autre part. L'ensemble des résultats principaux de notre recherche peut être synthétisé de la façon suivante.

L'importance croissante des logiciels

L'importance croissante des logiciels dans la vie économique et sociale – que l'on peut qualifier de *logicialisation des activités* – a été démontrée à un double niveau.

Tout d'abord, il apparaît nettement que l'information et les connaissances jouent un rôle croissant dans l'économie, que ce soit au niveau de l'importance prise par les biens et services informationnels, ou au niveau du fonctionnement des organisations comme de l'ensemble de l'économie. Les technologies de l'information et de la communication, technologies "invasives", ont joué un rôle majeur dans cette évolution. Par contre, elles ne nous semblent pas conduire à l'existence d'un nouveau paradigme unique, même qualifié de socio-économique, en raison de la diversité des évolutions dont elles sont porteuses et de la pluralité durable des configurations productives, représentant différentes cohérences possibles entre changements socio-organisationnels, changements techniques et changements économiques. Au sein de ces technologies de l'information et de la communication en forte croissance, l'extension du rôle des logiciels est un phénomène incontestable et qui va se poursuivre, malgré une sous-estimation durable de leur importance, et en dépit des difficultés d'évaluation statistique.

Au niveau plus précisément de l'informatique, l'analyse de son évolution est également riche d'enseignements. Nous avons analysé l'histoire de l'informatique comme la constitution successive de réseaux technico-économiques structurés par des standards. Nous avons vu que les processus de standardisation pouvaient être l'œuvre d'acteurs divers (entreprises, Etat mais également organismes alternatifs), qu'ils pouvaient ne pas être nécessairement optimaux et malgré tout perdurer, permettant la consolidation et l'extension d'un réseau technico-économique, qui en retour renforce ces standards. Toutefois, dans un contexte marqué par la très forte croissance de l'informatique, l'existence d'innovations techniques et de nouveaux besoins, correspondant notamment à des changements culturels, a permis l'émergence de nouveaux réseaux technico-économiques, qui coexistent avec les anciens, la situation actuelle étant marquée par une fusion tendancielle entre ces différents réseaux. Ce que montre cette histoire, c'est que les problèmes déterminants (notamment les questions de standardisation) se sont déplacés progressivement du champ du matériel informatique vers le champ des logiciels, et que les acteurs majeurs de l'informatique sont de moins en moins des producteurs de matériels et de plus en plus des producteurs de logiciels.

Les spécificités du logiciel

Dans le traitement automatique de l'information, le logiciel a un statut original d'objet frontière entre le monde humain et le monde machinique, "d'acteur non humain" partie prenante d'un système informatique. Un logiciel se présente sous la forme d'un texte numérique actif doté de nombreuses caractéristiques, qui sont le reflet de la diversité des points de vue existants.

Le logiciel apparaît comme un objet technologiquement complexe : il vise à apporter des solutions à des problèmes de plus en plus complexes – que dans certains cas il contribue à complexifier davantage – en pilotant un système, le système informatique, qui se caractérise également par une grande complexité. Il existe une dynamique puissante d'augmentation simultanée de la puissance des matériels et de la complexité des logiciels, basée sur la force des relations de complémentarité techniques et économiques entre ces deux activités.

Malgré sa jeunesse, le logiciel est marqué par un foisonnement d'innovations, qui concernent l'extension accélérée de ses domaines d'applications, la création de nouveaux produits, le renouvellement des méthodes et des techniques de production, et l'utilisation d'outils d'automatisation (génie logiciel). La forte dynamique innovatrice de l'économie du logiciel est confirmée par le fait que les innovations dans le domaine des logiciels concernent *tous* les types d'innovations mis en évidence par les économistes qui souhaitent appréhender l'ensemble des innovations existant dans l'économie. Par contre, les rythmes de diffusion très différenciés de ces innovations expliquent l'hétérogénéité technologique de l'économie du logiciel et le fait qu'elle semble caractérisée par une instabilité technologique permanente.

Sur un plan plus économique, le développement de logiciels sur mesure constitue une activité de services, alors que les progiciels sont des biens intangibles. Les progiciels possèdent "naturellement" des caractéristiques de biens collectifs des biens intangibles. De ce fait, pour rendre un progiciel appropriable, il est nécessaire de recourir à des dispositifs techniques (à efficacité limitée) et à des dispositifs juridiques. La variété de ces dispositifs, les problèmes que pose leur application aux progiciels, expliquent la diversité des statuts juridiques des progiciels et l'apparition de formes originales comme les licences publiques à la base des logiciels libres. Enfin, les logiciels se caractérisent par des mécanismes singuliers de détermination de leur prix. Les théories de la valeur ne semblent pas pouvoir s'appliquer aisément au cas des logiciels : il est particulièrement difficile d'appréhender la valeur d'usage d'un logiciel, ce qu'illustre particulièrement les débats sur l'appréciation des effets de

l'informatique ("paradoxe de Solow"). Quant à la théorie de la valeur travail, elle ne peut s'appliquer à des biens qui, soit ne sont pas reproductibles (logiciels sur mesure), soit sont trop facilement reproductibles (progiciels).

Pluralité et dynamique des mondes de production des logiciels

Indépendamment des progrès bien réels existants, la production des logiciels présente des spécificités qui rend particulièrement difficile la réalisation de logiciels qui seraient conjointement très fiables, bien adaptés aux besoins des utilisateurs et produits avec de hauts niveaux de productivité. En particulier la résolution d'une des questions critiques de l'économie du logiciel rend souvent impossible des progrès identiques sur les autres dimensions. Il en résulte que dans l'économie du logiciel se développent simultanément des processus différents de rationalisation de l'activité, distincts selon les nécessaires compromis opérés entre les différentes caractéristiques de l'activité, ce qui confère à l'économie du logiciel une grande diversité.

Cette diversité a été analysée en termes de mondes de production des logiciels. Chacun des mondes de production occupe une position particulière par rapport aux questions critiques de la production des logiciels. Le *monde interpersonnel* représente les difficultés à concilier productivité et fiabilité élevées dans la production de logiciels sur-mesure. Le *monde fordiste* des progiciels commerciaux se caractérise par une forte productivité mais une qualité souvent décevante, et ne peut répondre qu'à des besoins standard. Le *monde de la création* des logiciels libres apporte une certaine originalité et une efficacité indéniable sur certains segments de l'économie des logiciels, mais rencontre plus de difficultés pour s'étendre à l'ensemble des utilisateurs et des produits. Le *monde de la production flexible* des logiciels est porteur de potentialités pour concilier productivité et adaptation aux besoins des utilisateurs, mais il a du mal à se développer, surtout dans sa composante la plus prometteuse (la production de logiciels sur mesure à partir de composants standard).

Si la coexistence de mondes de production différents est un phénomène durable dans l'économie du logiciel, notamment en raison des très fortes complémentarités existant entre ces mondes, il n'empêche que l'économie du logiciel connaît d'importantes évolutions dynamiques. Ces changements se traduisent par l'existence de *plusieurs* trajectoires, correspondant au déplacement d'une production effectuée dans un monde de production vers un autre monde de production. A côté d'une trajectoire d'industrialisation, qui aboutit à une production en grande série de biens standard (monde "fordiste spécifique" des progiciels

commerciaux), existent également des trajectoires de valorisation, de flexibilisation et d'objectivation, qui peuvent permettre un développement du monde de production flexible des logiciels, à partir des trois autres mondes de production. Cette analyse a permis de mettre en évidence la domination *actuelle* des Etats-Unis sur l'économie du logiciel, grâce à une industrialisation beaucoup plus rapide de leur production. Mais l'existence d'autres trajectoires, et notamment de celle que nous avons qualifiée de valorisation, peut contribuer à modifier cette situation. De ce point de vue, le développement du monde de production des logiciels libres, outre sa contribution au nécessaire maintien d'une certaine diversité des progiciels, peut être riche de potentialités pour favoriser l'avènement d'un véritable "sur mesure de masse" dans l'économie du logiciel, grâce à la production de composants logiciels libres. Toutefois, il n'est pas évident que l'extension du champ des logiciels libres se poursuive aussi spontanément que cela a été le cas jusqu'à présent. L'analyse des différents facteurs du développement des logiciels libres montre que les facteurs défavorables à leur essor risquent de dominer les facteurs favorables. De ce point de vue, une action volontariste et multiforme des pouvoirs publics en faveur des logiciels libres pourrait être bénéfique.

Cette recherche pourrait être approfondie, notamment en ce qui concerne l'étude des dynamiques de cette économie. Une poursuite des investigations empiriques sur une période plus longue (certains phénomènes comme le développement du monde de la création, et certaines modalités du monde de production flexible sont très récents) et la construction d'un cadre théorique complet et adapté, qui intégrerait notamment certains apports des théories évolutionnistes, pourrait permettre de valider ou d'infirmer les hypothèses émises.

L'utilisation d'un tel cadre théorique pourrait être également mobilisée pour effectuer des comparaisons sur les dynamiques que connaissent d'autres secteurs économiques, également marqués par la pluralité des formes de coordination de l'activité. A terme, un objectif très ambitieux, mais qui nous semble non dénué d'intérêt, serait d'intégrer l'étude des dynamiques sectorielles et l'étude de la dynamique d'ensemble d'une économie, dans la mesure où il existe des interdépendances fortes entre ces deux niveaux d'analyses. Cette analyse pourrait utilement s'appuyer sur certains travaux récents des économistes de l'école de la régulation.

De façon peut-être plus réaliste à court terme, un autre axe de recherche pourrait être d'utiliser certains résultats de notre recherche pour analyser les modalités des échanges qui se déroulent par l'intermédiaire d'Internet. Ces échanges se développent rapidement sous de multiples formes : *Business To Business*, *Business To Consumer* dans une moindre mesure, mais aussi *Peer to Peer*. Le développement de ces échanges "d'égal à égal" constitue d'une certaine façon un retour à la philosophie initiale d'Internet, mais pose de redoutables défis économiques et juridiques. De façon plus générale, l'hypothèse que nous pourrions alors tester est qu'il nous semble peu vraisemblable que les échanges économiques réalisés par l'intermédiaire d'Internet s'effectueront selon un modèle dominant qui s'imposerait inéluctablement, et qu'au contraire coexisteront durablement plusieurs façons de penser la production au sens large, dont certaines intégreront les principales caractéristiques des mondes de production des logiciels.

Nous espérons que l'analyse de l'économie du logiciel que nous avons effectuée a permis d'enrichir la compréhension de cette activité. L'étude de ce secteur, et notamment la place qu'a prise le monde de la création des logiciels libres, nous a conforté dans la nécessité d'intégrer d'autres analyses du comportement humain que la seule rationalité instrumentale, et d'autres mécanismes de coordination que le seul recours au marché. L'étude des perspectives d'évolution de l'économie des logiciels intègre quelques propositions d'action des pouvoirs publics. Une satisfaction personnelle serait que ce travail contribue modestement à une réflexion sur une "informatique au service du plus grand nombre et non accaparée pour les plus grands profits du plus petit nombre" (Roberto Di Cosmo, Dominique Nora, 1998, p. 12).

RÉFÉRENCES BIBLIOGRAPHIQUES

- ABERNATHY W., UTTERBACH J. (1978), « Patterns of industrial innovation », in *Technology Review*, n° 80, juin-juillet, pp. 41-47.
- ABRAMOWITZ M., DAVID P.A. (1996), « Technological change and the rise of intangible investments : the US economy's growth-path in the twentieth century », in FORAY Dominique, LUNDVALL Bengt Ake (Eds.) *Employment and growth in the knowledge-based Economy*, OCDE, 389 p.
- AFRIAT Christine (1992), *L'investissement dans l'intelligence*, PUF (Que sais-je?), 128 p.
- AFRIAT Christine, CASPAR Pierre (1988), *L'investissement intellectuel : Essai sur l'économie immatérielle*, Economica, 184 p.
- AGRE Philip E. (1997), « Surveillance et saisie : Deux modèles de l'information personnelle », pp. 243-266, in CONEIN Bernard, THEVENOT Laurent (eds.), *Cognition et information en société*, EHESS (Raisons pratiques 8), 317 p.
- AKERLOF George A. (1970), « The Market for "Lemons" : Quality, Uncertainty and the Market Mechanism », in *Quarterly Journal of Economics*, 84(3), August, pp. 488-500.
- ALBRECHT A.J., GAFFNEY J.E. (1983), « Software Function, Source Lines of Code, and Development Effort Prediction », in *IEEE Transactions on Software Engineering*, Vol. SE-9, n° 6, pp. 639-647.
- ANDERSON Alan Ross (1983), *Pensée et machine*, Champ Vallon (Collection Milieux), 150 p.
- ANGELIER Jean-Pierre (1997), *Economie industrielle: une méthode d'analyse sectorielle*, Presses Universitaires de Grenoble (L'économie en plus), 158 p.
- AOKI Masahiko (1986), « Horizontal Vs Vertical Information Structure of the Firm », in *American Economic Review*, n° 76, pp. 971-983.
- AOKI Masahiko (1991), « Le management japonais : le modèle 'J' d'Aoki », in *Problèmes économiques*, n° 2225.
- ARCANGELI Fabio, GENTHON Christian (1997), « Gestion et partage à distance de la connaissance tacite : un défi pour le multimédia », pp. 155-166, in MONNOYER-LONGE Marie-Christine (éd.), *L'entreprise et l'outil informationnel*, L'Harmattan (Communication), 220 p.
- ARCHIMBAUD Jean-Luc (1995), « Comprendre l'Internet », pp. 17-40, in *L'Internet professionnel*, CNRS Ed. (LMB Monographie), 448 p.
- ARENA Richard (et al.) (1991), *Traité d'économie industrielle*, Economica, 1001 p.
- ARGYRIS Chris, SCHÖN Donald A. (1978), *Organisational Learning : a Theory of Action Perspective*, Addison-Wesley, 344 p.
- ARON Patrice (1998), « Logiciel libre : la subversion gagne », in *Le Monde Informatique*, n° 774.
- ARROW Kenneth J. (1962), « Economic welfare and the allocation of resources for invention », pp. 609-626, in *The Rate and direction of inventive activity : economic and social factors*, Princeton University Press, 635 p.
- ARSAC Jacques (1987), *Les machines à penser : des ordinateurs et des hommes*, Seuil (Science ouverte), 250 p.
- ARTHUR Brian (1988), « Competing Technologies : An Overview », in DOSI Giovanni (et al.), *Technical change and economic theory*, Pinter, (IFIAS research series), 646 p.
- ARTHUR Brian (1989), « Competing Technologies, Increasing Returns and Lock-in by Historical Events », in *Economic Journal*, vol. 99, pp. 106-131.
- ASSOCIATION NATIONALE DE LA RECHERCHE TECHNIQUE (1999), *L'innovation dans les services : une invitation à l'insurrection intellectuelle*, Economica, 98 p.
- AURAY Nicolas (1997), « Ironie et solidarité dans un milieu technicisé. Les défis contre les protections dans les collectifs de hackers », in CONEIN Bernard, THEVENOT Laurent (eds.), *Cognition et information en société*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Raisons pratiques 8), 317 p.
- BABELON Christine (1987), *L'Europe et le marché mondial du logiciel*, Thèse de Troisième Cycle, Université Grenoble II.

- BADILLO Patrick-Yves (1996), « Organisation de marché : 'slack' technologique et réglementaire. Analyse théorique et application au secteur de la communication », in BROUSSEAU Eric, PETIT Pascal, PHAN Denis (eds.), *Mutation des Télécommunications, des Industries et des Marchés*, Economica, ENSPTT (Management communication réseaux).
- BARBROOK Richard (2000), « La liberté de l'hypermédia », in BLONDEAU Olivier, LATRIVE Florent (2000), *Libres enfants du savoir numérique*, L'Eclat, 502 p.
- BARCET André, BONAMY Joël, MAYERE Anne (1984), « Les services aux entreprises : Problèmes théoriques et méthodologiques », in *L'Economie des services : Recherches économiques et sociales*, La Documentation française.
- BARCET André, BONAMY Joël, MAYERE Anne (1987), *La qualité des services : de l'analyse économique aux processus institutionnels de qualification des services intellectuels*, Ministère de la Recherche et de la Technologie, septembre.
- BASLE Maurice (et al.) (1995), *Changement institutionnel et changement technologique : évaluation, droits de propriété intellectuelle, système national d'innovation*, CNRS Editions, 333 p.
- BAUMOL William J. (1967), « Macroeconomics of Unbalanced Growth : the anatomy of urban Crisis », in *American Economic Review*, juin, pp. 415-426.
- BAUMOL William J. (1982), « Contestable Markets : An Uprising in the Theory of Industrial Structure », in *American Economic Review*, march.
- BAUMOL William J., BLACKMAN Sue Anne Batey, WOLF Edward N. (1985), « Unbalanced growth revisited : asymptotic stagnancy and new evidence », in *American Economic Review*, vol. 75, n° 4, pp. 332-337, traduit sous le titre « Nouvel examen du modèle de croissance déséquilibrée : application au cas américain », in *Problèmes Economiques*, n° 1970, 16 avril 1986, pp. 14-22.
- BELL Daniel (1976), *Vers la société post-industrielle*, trad. de *The Coming of post-industrial society*, R. Laffont, 446 p.
- BELLEFLAMME Claudine (et al.) (1986), *Innovation and Research and Development Process Analysis in Service Activities*, IRES, rapport FAST-CEE.
- BELLON Bruno (1991), « La compétitivité », pp. 482-497, in ARENA Richard (et al.), *Traité d'économie industrielle*, Economica, 1001 p.
- BERTHIER Jean-Pierre (1999), « Les nouvelles évaluations de biens et services dans les comptes nationaux », in *Economie et Statistique*, INSEE, n° 321-322, pp. 7-31.
- BILLOTTE Pierre-José (1997), *Concurrence technologique et normalisation : Enjeux publics et stratégies industrielles*, AFNOR, 222 p.
- BIRRIEN Jean-Yvon (1990), *Histoire de l'informatique*, PUF (Que sais-je?), 127 p.
- BLONDEAU Olivier, LATRIVE Florent (2000), *Libres enfants du savoir numérique*, L'Eclat, 502 p.
- BOASSON Maarten (1998), « Quelques mouvements de plus pour l'impossible », in *Génie logiciel*, n° 47, mars, pp. 2-10.
- BOEHM Barry W. (1981), *Software engineering economics*, Prentice-Hall, 767 p.
- BOLTANSKI Luc, THEVENOT Laurent (1997), *De la justification : Les économies de la grandeur*, Gallimard (Les Essais), 483 p.
- BONNAURE Pierre (1996), « L'aventure de B. Gates et de Microsoft. A propos de la route du futur », in *Futuribles*, avril, pp. 33-37.
- BONNAURE Pierre (1996), « L'informatique dans l'impasse ? : La 3e vague à la croisée des chemins », in *Futuribles*, novembre, pp. 65-71.
- BOUCHY Serge (1994), *L'ingénierie des systèmes d'information évolutifs*, Eyrolles, 329 p.
- BOULDING Kenneth E. (1988), *Information: the source and the enemy of knowledge*, School of Library and Information Management, State University, 17 p.
- BOWER J.L., CHRISTENSEN C.M. (1995), « Disruptive technologies : catching the wave », in *Harvard Business Review*, 73 (1), pp. 43-53.

- BOWKER Geoffrey, STAR Susan Leigh (1997), « Problèmes de classification et de codage dans la gestion internationale de l'information », pp. 283-310, in CONEIN Bernard, THEVENOT Laurent (eds.), *Cognition et information en société*, EHESS (Raisons pratiques 8), 317 p.
- BOYER Robert (1993), *Une contribution au renouveau d'une économie institutionnaliste : la théorie de la régulation dans les années 1990*, Cepremap, 30 p.
- BOYER Robert, CHAVANCE Bernard, GODARD Olivier (eds.) (1991), *Les figures de l'irréversibilité en économie*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Recherches d'histoire et de sciences sociales), 393 p.
- BOYER Robert, ORLEAN André (1991), « Les transformations des conventions salariales entre théorie et histoire », in *Revue Economique*, 42 (2), pp. 233-272.
- BOYER Robert, ORLEAN André (1994), « Persistance et changement des conventions. Deux modèles simples et quelques illustrations », pp. 219-248, in ORLEAN, André (éd.), *Analyse économique des conventions*, Presses Universitaires de France (Economie), 403 p.
- BOYER Robert, SAILLARD Yves (1995), « Un précis de régulation », pp. 55-68, in BOYER Robert, SAILLARD Yves (eds.), *Théorie de la régulation : l'état du savoir*, La Découverte (Recherches), 568 p.
- BRAMAN Sandra (1997), « L'économie de l'information : une évolution des approches », pp. 87-111, in MAYERE Anne (éd.), *La société informationnelle : enjeux sociaux et approches économiques*, L'Harmattan (Communication et civilisation), 239 p.
- BRES Jean (1994), *Ateliers de génie logiciel : Réalités et tendances*, Masson (MIPS), 253 p.
- BRESSAND Albert (1994), « Réseaux et marchés-réseaux », pp. 263-281, in DE BANDT Jacques, GADREY Jean (eds.), *Relations de service, marchés de service*, CNRS ed (Recherche et entreprise), 360 p.
- BRESSAND Albert, NICOLAIDIS Kalypso (1988), « Les services au cœur de l'économie relationnelle », in *Revue d'Economie Industrielle* n°43.
- BRETON Philippe (1987), *Histoire de l'informatique*, La découverte, 239 p.
- BRETON Philippe (1990), *La tribu informatique : Enquête sur une passion moderne*, Métailié, (collection Traversées), 190 p.
- BREVIGLIERI Marc (1997), « La coopération spontanée : Entraides techniques autour d'un automate public », pp. 123-148, in CONEIN Bernard, THEVENOT Laurent (éd.), *Cognition et information en société*, Editions de l'école des Hautes Etudes en Sciences Sociales (Raisons pratiques 8), 317 p.
- BRINBERG Herbert R., PINELLI Thomas E. et BARCLAY Rebecca O. (1997), « Déterminer la valeur de l'information dans un environnement interactif », pp. 181-196, in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- BRISSET Antoine (1999), *Mémoire en marketing sur le marché du système d'exploitation Linux*, IAE Paris.
- BROOKS Frederick P. (1996), *Le mythe du mois-homme : Essais sur le génie logiciel*, International Thomson Publishing, 276 p., traduction de *The Mythical Man-Month*, Addison-Wesley Publishing Company, (1995).
- BROUSSEAU Eric (1993), *L'Economie des contrats : Technologies de l'information et coordination interentreprises*, P.U.F. (Economie en liberté), 368 p.
- BROUSSEAU Eric (1997), « Technologies de l'information et de la communication, gains de productivité et transformation des dispositifs de coordination », pp. 41-65, in MONNOYER-LONGE Marie-Christine (éd.), *L'entreprise et l'outil informationnel*, L'Harmattan (Communication), 220 p.
- BROUSSEAU Eric, PETIT Pascal, PHAN Denis (eds.) (1996), *Mutation des télécommunications, des industries et des marchés*, Economica, ENSPTT (Management, communication réseaux), 553 p.
- BROUSSEAU Eric, RALLET Alain (1997), « Le rôle des technologies de l'information et de la communication dans les changements organisationnels », pp. 286-309, in GUILHON Bernard (et al.), *Economie de la connaissance et organisations : entreprises, territoires, réseaux*, L'Harmattan, 481 p.

- BROUSSEAU Eric., GEOFFRON Patrice, WEINSTEIN Olivier (1997), « Confiance, connaissances et relations inter-firmes », pp. 402-433, in GUILHON Bernard (et al.), *Economie de la connaissance et organisations : entreprises, territoires, réseaux*, L'Harmattan, 481 p.
- BROWN Ronald H. (1995), *The Global Information Infrastructure : agenda for cooperation*, U.S. G.P.O, 49 p.
- BRULE Jean-Pierre (1993), *L'informatique malade de l'Etat : Du plan calcul à Bull nationalisée, un fiasco de 40 milliards*, Les Belles Lettres (Témoignages et documents), 380 p.
- BRYNJOLFSSON Erik (1993), « The productivity paradox of information technology », in *Communications of ACM*, Dec, Vol. 36, n° 12, pp. 67-77.
- BURAWOY Michael (1985), *The Politics of Production*, Verso, 272 p.
- BUREAU OF ECONOMIC ANALYSIS (1997), *Estimation of investment in computer software*, US Department of Commerce, 4 p.
- BUSINESS SOFTWARE ALLIANCE (1999), *Forecasting a robust future*, 26 p.
- CACCOMO Jean-Louis (1996), *Les défis économiques de l'information : la numérisation*, L'Harmattan (Economie et innovation), 126 p.
- CALLENS Stéphane (1995), « De la prospection à la relation de suivi », in *Cahiers du Centre d'Etudes de l'Emploi* n°34, pp. 119-144.
- CALLON Michel (éd.) (1989), *La science et ses réseaux : genèse et circulation des faits scientifiques*, La Découverte, UNESCO, Conseil de l'Europe (Textes à l'appui), 214 p.
- CALLON Michel (1991), « Réseaux technico-économiques et irréversibilités », pp. 195-230, in BOYER Robert, CHAVANCE Bernard, GODARD Olivier (eds.), *Les figures de l'irréversibilité en économie*, Edition de l'Ecole des Hautes Etudes en Sciences Sociales, 393 p.
- CALLON Michel (1992), « Variété et irréversibilité dans les réseaux de conception et d'adoption des techniques » pp. 275-324, in FORAY Dominique, FREEMAN Christopher (eds), *Technologie et richesse des nations*, Economica, 517 p.
- CALLON Michel (1993), *La privatisation de la science est-elle inéluctable ?*, Ronéo, 26 p., version française légèrement différente de Michel CALLON (1994 B).
- CALLON Michel (1994 A), « L'innovation technologique et ses mythes », in *Gérer et Comprendre*, mars, pp.5-17.
- CALLON Michel (1994 B), « Is Science a Public Good ? », in *Science, Technology and Human Value*, vol. 19.
- CALLON Michel (1999), « L'importance du tertiaire dans l'innovation », pp. 19-23, in ASSOCIATION NATIONALE DE LA RECHERCHE TECHNIQUE (1999), *L'innovation dans les services : une invitation à l'insurrection intellectuelle*, Economica, 98 p.
- CALLON Michel, LATOUR Bruno (eds.) (1991), *La science telle qu'elle se fait*, La Découverte (Textes à l'appui), 390 p.
- CARROLL Paul (1994), *Big Blues chez IBM ou le déclin d'un empire américain*, Addison-Wesley, 378 p.
- CASTELLS Manuel (1998), *La société en réseaux : l'ère de l'information*, Fayard, 613 p.
- CASTELLS M., AOYOMA Y. (1994), « Vers la société de l'information : structures de l'emploi dans les pays du G7 », in *Revue internationale du travail*, vol. 133, n° 1.
- CATINAT Michel (1998), « La politique communautaire de stimulation de la société de l'information », pp.37-52, in PETIT Pascal (éd.) (1998), *L'économie de l'information : les enseignements des théories économiques*, La Découverte (Recherches), 405 p.
- CEZARD M., DUSSERT F., GOLLAC M. (1993), « Taylor va au marché. Organisation du travail et informatique », in *Travail et Emploi* n° 54.
- CHAMBERLIN Edward H. (1933), *The Theory of Monopolistic Competition*, Harvard University Press.
- CHANDLER Alfred D.(1992), *Organisation et performance des entreprises (Scale and scope 1990), Tome 1 - Les USA 1880-1948*, Editions d'organisation, 429 p.
- CHARBIT Claire, ZIMMERMANN Jean-Benoît (1997), « Systèmes d'information et entreprises : convergence ou incertitude ? », in *Terminal, Technologie de l'Information, Culture et Société*, n°78.
- CIBORRA Claudio (1993), *Teams, markets and systems*, Cambridge University Press, 250 p.

- CLOT-LAFLEUR Ghislaine, PASSERON Hervé (1997), « Technologie de l'information et emploi en France », in *Communications et Stratégies*, n° 28, pp. 129-148.
- CLUB DE L'ARCHE (1993), « L'aube d'un monde nouveau ? Vers une "numérisation" de la société ? », in *Problèmes Economiques*, n° 2464-2465, pp. 3-5.
- COASE Ronald H. (1937), « The nature of the Firm », in *Economica*, vol. 16, pp. 331-351, traduction française, « La nature de la firme », in *Revue Française d'Economie*, vol. II, n° 1, hiver 1987, pp. 133-157.
- COHENDET Patrick (1996), « Réseaux, Pertinence et Cohérence » pp. 137-158, in COHENDET Patrick, JACOT Jacques-Henri, LORINO Philippe, *Cohérence, pertinence et évaluation*, *Economica*, 308 p.
- COHENDET Patrick (1997), « Information, connaissances et théorie de la firme évolutionniste », pp. 98-110, in GUILHON Bernard (et al.), *Economie de la connaissance et organisations : entreprises, territoires, réseaux*, L'Harmattan, 481 p.
- COLLINS Harry M. (1992), *Experts artificiels : Machines intelligentes et savoir social*, Seuil (Science ouverte), 310 p.
- COMMISSARIAT GENERAL DU PLAN (1990), *Information et compétitivité* (rapport du groupe présidé par René Mayer), La Documentation Française, 302 p.
- COMMISSARIAT GENERAL DU PLAN (1992), *L'informatique de l'Etat, rapport à la demande du Comité Interministériel de l'Evaluation des Politiques Publiques* (CIME), La Documentation Française, 280 p.
- COMMISSION DES COMMUNAUTES EUROPEENNES (1991), *Panorama de l'industrie communautaire 1991-92 : situation actuelle et perspectives pour 180 secteurs des industries manufacturières et de services dans la C.E.*, Office de publications officielles des communautés européennes.
- CONEIN Bernard (1997), « L'action avec les objets : Un autre visage de l'action située ? », pp. 25-46, in CONEIN Bernard, THEVENOT Laurent (eds.), *Cognition et information en société*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Raisons pratiques 8), 317 p.
- CONEIN Bernard, DODIER Nicolas et THEVENOT Laurent (eds.) (1993), *Les objets dans l'action. De la maison au laboratoire*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Raisons pratiques 4), 290 p.
- CONEIN Bernard, THEVENOT Laurent (eds.) (1997), *Cognition et information en société*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales, (Raisons pratiques 8), 317 p.
- CONFERENCE MINISTERIELLE DU G7 (1995), *Une vision partagée de l'enrichissement humain*, conférence interministérielle du G7 sur la société de l'information.
- COOKE Philip (et al.) (1992), *Towards global localization : The computing and telecommunications industries in Britain & France*, UCL Press, 227 p.
- CORIAT Benjamin (1994), « La théorie de la régulation. Origines, spécificités et perspectives », in *Futur Antérieur*, novembre-décembre, pp.101-152.
- CORIAT Benjamin, TADDEI Dominique (1993), *Made in France 2 : entreprise France*, Librairie générale française (Le livre de poche), 416 p.
- CORIAT Benjamin, WEINSTEIN Olivier (1995), *Les nouvelles théories de l'entreprise*, Librairie générale française (Le livre de poche), 218 p.
- COTTE Dominique (1999), *Sauts technologiques et ruptures dans les modes de représentation des connaissances. Etude du texte numérique comme objet technique*, Thèse de Sciences de l'Information et de la Communication, Université Lille 3, 291 p.
- COUTROT Thomas (1998), *L'entreprise néolibérale, nouvelle utopie capitaliste ? Enquête sur les modes d'organisation du travail*, La Découverte (Textes à l'appui), 281 p.
- COWAN Robin (1995), « L'informatisation des gouvernements : enjeux économiques et choix technologiques », in *STI Revue* n° 16, OCDE, pp. 217-251.
- COWAN Robin, FORAY Dominique(1998), « Economie de la codification et de la diffusion des connaissances », pp.301-329, in PETIT Pascal (éd.), *L'économie de l'information : les enseignements des théories économiques*, La Découverte (Recherches), 405 p.

- CRINGELY Robert X. (1993), *Bâtisseurs d'empires par accident : origines et dessous de la Silicon Valley*, Addison-Wesley, 307 p.
- CRONIN Blaise (1997), « La société informationnelle : enjeux sociaux et approches économiques », pp. 9-17, in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- CROZET Yves (1997), *Analyse économique de l'Etat*, A. Colin (Cursus économie), 191 p.
- CROZIER Michel (1963), *Le phénomène bureaucratique, essai sur les tendances bureaucratiques des systèmes d'organisation modernes et sur leurs relations en France avec le système social et culturel*, Seuil, 413 p.
- CROZIER Michel (1971), *La société bloquée*, Seuil (Points Politique), 251 p.
- CROZIER Michel, FRIEDBERG Ehrard (1988), *L'acteur et le système : les contraintes de l'action collective*, Seuil (Sociologie politique), 436 p.
- CURIEN Nicolas (1993), « Economie des services en réseau : principes et méthodes », in *Communications et Stratégies* n° 10, p. 13-30.
- CUSUMANO Michael A. (1991), *Japan's software factories : A challenge to U.S. Management*, Oxford University Press, 513 p.
- CUSUMANO Michael A., SELBY Richard W. (1996), *Microsoft Secrets : How the world's most powerful software company creates technology, shapes markets, and manages people*, HarperCollins, 512 p.
- CXP (1996), *Les catalogues de progiciels : génie logiciel*, CXP.
- DAFT R.L., MAC INTOSH N.B. (1978), « A New Approach to design and Use of Management Information », in *California Management Review*, Autumn, pp. 82-92.
- DALLE Jean-Michel (1995), « Dynamiques d'adoption, coordination et diversité : la diffusion des standards technologiques », in *Revue économique* vol. 46, n° 4, pp. 1081-1098.
- DALLE Jean-Michel, JULLIEN Nicolas (2000), *Windows vs. Linux : une investigation du système économique du logiciel libre*, Ronéo, 21 p.
- DANG N'GUYEN Godefroy (1995), *Economie industrielle appliquée*, Vuibert (Economie), 373 p.
- DANG N'GUYEN Godefroy, PENARD Thierry (1998), « Les accords d'interconnexion dans l'Internet : enjeux économiques et perspectives réglementaires », in *Communications et Stratégies* n° 32, pp. 107-133.
- DANG N'GUYEN Godefroy, PENARD Thierry (1999), « Don et coopération dans Internet : une nouvelle organisation économique », in revue *Terminal, Technologie de l'Information, Culture et Société*, n° 80/81, « Les logiciels libres : de l'utopie au marché », pp. 95-116.
- DANG N'GUYEN Godefroy, PETIT Pascal, PHAN Denis (1997), « Les enjeux économiques et sociaux de la société de l'information », in *Communications et Stratégies* n° 28, pp. 33-86.
- DANG NGUYEN Godefroy, PHAN Denis (1997), « Apprentissages et diffusion du paradigme 'numérique' dans les Technologies de l'Information et de la Communication », pp. 177-207, in GUILHON Bernard (et al.), *Economie de la connaissance et organisations : Entreprises, Territoires, Réseaux*, L'Harmattan, 481 p.
- DARES - Ministère de l'emploi et de la solidarité (1998), *L'emploi dans le secteur des technologies de l'information et des télécommunications*, Premières synthèses, 12 p.
- DASGUPTA Partha, DAVID Paul A. (1994), « Towards a new economics of science », in *Research Policy*, vol. 23-5.
- DASGUPTA Partha, STIGLITZ Joseph (1980), « Industrial structure and the nature of innovative activity », in *The Economic Journal*, 90.
- DAVENPORT Paul (1997), « Le paradoxe de la productivité et la gestion des technologies de l'information », pp. 9-19, in MONNOYER-LONGE Marie-Christine (éd.), *L'entreprise et l'outil informationnel*, L'Harmattan (Communication), 220 p.
- DAVID Paul A. (1985), « Clio and the Economics of QWERTY », in *American Economic Review*, 75, 2, pp. 332-337.
- DAVID Paul A. (1987), « Some New Standards for the Economics of Standardization in the Information Age », in DASGUPTA P., STONEMAN P., *Economic Policy and Technological Performance*, Cambridge U.P., 243 p.

- DAVID Paul A. (1991), « The Computer and the Dynamo : the Modern Productivity Paradox in a Not-Too-Distant Mirror », in *Technology and Productivity : The Challenge for Economic Policy*, OCDE, 588 p.
- DAVID Paul A. (1994), « Les standards des technologies de l'information, les normes de communication et l'Etat : un problème de biens publics », pp. 249-280, in ORLEAN André (éd.) (1994), *Analyse économique des conventions*, PUF (Economie), 403 p.
- DAVID Paul A. et BUNN J. (1988), « The economics of gateway technologies and network evolution », in *Information Economics and Policy*, 3.
- DAVID Paul A., FORAY Dominique (1995), « Distribution et expansion de la base de connaissances scientifiques et technologiques », in *STI Revue* n° 16, OCDE, pp. 13-73.
- DE BANDT Jacques (éd.) (1991), *Les services : Productivité et prix*, Economica, 264 p.
- DE BANDT Jacques (1995), *Services aux entreprises : Informations, produits, richesses*, Economica, 268 p.
- DE BANDT Jacques, GADREY Jean (eds.) (1994), *Relations de service, marchés de service*, CNRS Editions, (Recherches et entreprise), 360 p.
- DE BANDT Jacques, GADREY Jean (1994), « De l'économie des services à l'économie des relations de service », pp. 11-22., in DE BANDT Jacques, GADREY Jean (eds.)
- DE BESSES Marie (1999), « Les logiciels libres, un système de circulation fertile en apprentissages », in revue *Terminal, Technologie de l'Information, Culture et Société*, n° 80/81, « Les logiciels libres : de l'utopie au marché », pp. 257-278.
- DELAPIERRE Michel, ZIMMERMANN Jean-Benoît (1984), « Les multinationales de l'électronique : des stratégies différenciées », in *Revue d'Economie Industrielle* n° 28, pp. 9-35.
- DELAPIERRE Michel, ZIMMERMANN Jean-Benoît (1991), *La globalisation de l'industrie des ordinateurs*, rapport pour le programme Monitor-Fast de la Commission des Communautés Européennes, volume 11, dossier prospectif n°2 (Globalisation de l'Economie et de la Technologie), 153 p.
- DELAPIERRE Michel, ZIMMERMANN Jean-Benoît (1994), « L'industrie informatique et la globalisation de l'industrie et de la technologie », in revue *Terminal, Technologie de l'Information, Culture et Société*, Automne 1994, pp. 87-104.
- DEMARCO Toran, LISTER Timothy (1991), *Les hommes de l'ordinateur : Les conditions de la productivité des équipes de projets informatiques*, Masson (Productivité de l'entreprise), 189 p., traduction de *Peopleware Productive Projects and Teams*, Dorset House Publishing Co. (1987).
- DESAINTQUENTIN Jean-Marie, SAUTEUR Bernard (1991), *L'informatique éclatée : tendances actuelles 1991-1993*, Masson (Stratégies et systèmes d'information), 341 p.
- DESAINTQUENTIN Jean-Marie, SAUTEUR Bernard (1996), *Les nouvelles clés de l'informatique*, Editions Synthèse Informatique, 295 p.
- DESREUMAUX Alain (1997), « Structures de l'entreprise », pp. 3147-3173, in Y. Simon & P. Joffre (eds.), *Encyclopédie de gestion*, Economica, 3 vols.
- DESSIMOZ Jean-Daniel (1997), « Estimation quantitative et évaluation économique des informations et des connaissances », pp. 215-232, in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- DETIENNE Françoise (1998), *Génie logiciel et psychologie de la programmation*, Hermes, 184 p.
- DI COSMO Roberto, NORA Dominique (1998), *Le hold-up planétaire : La face cachée de Microsoft*, Calmann-Lévy, 187 p.
- DIBONA, Chris, OCKMAN Sam, STONE Mark (1999), *Open sources : voices from the open source revolution*, O'Reilly & Associates, 272 p.
- DIMECH Philippe, TURPIN Etienne (1998), « Le réseau Internet est-il économiquement viable ? », in *Communications et Stratégies* n° 32, pp. 23-66.
- DJELLAL Faridah (1995), *Changement technique et conseil en technologie de l'information*, L'Harmattan (Logiques économiques), 239 p.
- DOBROV Gennady M. (1979), « La technologie en tant qu'organisation », in *Revue internationale des Sciences Sociales*, vol. XXVI.

- DOCKES Pierre (1990), « Formation et transferts des paradigmes socio-techniques », in *Revue Française d'Economie*, volume V, n° 4, pp. 29-82.
- DODIER Nicolas (1995), *Les hommes et les machines : la conscience collective dans les sociétés technicisées*, Métaillié (Leçons de choses), 384 p.
- DOSI Giovanni (1982), « Technological Paradigms and Technological Trajectories : a Suggested Interpretation of the Determinants and Direction of Technological Change », in *Research Policy*, vol. 11, pp. 147-162.
- DOSI Giovanni (1984), *Technical change and industrial transformation : the theory and application to the semiconductor industry*, Mac Millan, 338 p.
- DOSI Giovanni (1988), « Sources, Procedures, and Microeconomic Effects of Innovation », in *Journal of Economic Literature*, XXVI, pp. 1120-1171.
- DOSI Giovanni, ORSENIGO Luigi (1988), « Structure industrielle et évolution technologique », pp. 13-33, in HEERTJE Arnold (éd.), *Innovation, technologie et finance*, Basil Blackwell, 199 p.
- DREAN Gérard (1996 A), *L'industrie informatique : Structure, économie, perspectives*, Masson (Stratégie et systèmes d'information), 389 p.
- DREAN Gérard (1996 B), « Les inforoutes et l'industrie informatique », in *Réalités Industrielles*, Octobre - Novembre 1996.
- DROMBY Frédéric (1999), *L'inertie de la firme dominante face à l'innovation de produit : une explication par un phénomène d'information trompeuse en provenance du marché : L'exemple des logiciels d'application pour micro-ordinateurs*, Thèse de L'Ecole des Hautes Etudes Commerciales, 800 p.
- DRUCKER Peter Ferdinand (1993), *Post-Capitalist Society*, Harperbusiness, 232 p.
- DU GRANRUT Charles (1996), « Une brève histoire d'Internet », in *Futuribles*, novembre 1996, pp. 73-79.
- DUBUISSON Sophie (1997), « Stabilisation et réactualisation des connaissances sur la demande dans la dynamique productive : Quelques enseignements tirés d'une activité de service », pp. 310-335, in GUILHON Bernard, HUARD Pierre, ORILLARD Magali, ZIMMERMANN Jean-Benoît (eds), *Economie de la connaissance et organisations : Entreprises, Territoires, Réseaux*, L'Harmattan, 481 p.
- DUFOURT Daniel (1997), « Information, incertitude et comportements : pour une réévaluation des transactions relatives à l'information dans le contexte d'une économie de l'ignorance », pp. 149-168, in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- DUMOULIN Christiane (1986), *Management de système d'information*, les Editions d'organisation, 248 p.
- DUPUY Jean-Pierre (et al.)(1989), « Introduction à "L'économie des conventions" », in *Revue Economique*, vol. 40, n° 2, pp. 141-145.
- ETZIONI Amitai (1988), *The Moral Dimension : Toward a New Economics*, The Free Press, 314 p.
- EUROSTAF (1994 A), *Le marché français des services informatiques : l'avenir des SSII face aux nouveaux entrants*, Eurostaf (Perspectives stratégiques et financières), 325 p.
- EUROSTAF (1994 B), *Cap Gemini Sogeti*, Eurostaf, Série Stratégies et performances comparées des groupes, 66 p.
- EUROSTAF (1995 A), *Les matériels et services informatiques dans le monde*, Eurostaf, 71 p.
- EUROSTAF (1995 B), *Les matériels et services informatiques : note sectorielle*, Eurostaf, 74 p.
- EUROSTAF (1995 C), *L'infogérance (facilities management)*, Eurostaf (Dynamique des marchés), 52 p.
- EUROSTAF (1996 A), *L'industrie européenne des services informatiques. Tome 1 : Les marchés*, Eurostaf.
- EUROSTAF (1996 B), *L'industrie européenne des services informatiques. Tome 2 : Les intervenants*, Eurostaf, 274 p.
- EUROSTAF (1996 C), *Le marché mondial des logiciels*, Eurostaf, 321 p.
- EUROSTAF (1996 D), *Axime*, Eurostaf, Série Stratégies et performances comparées des groupes, 56 p.
- EUROSTAF (1996 E), *Sema Group*, Eurostaf, Série Stratégies et performances comparées des groupes, 88 p.
- EUROSTAF (1997 A), *Informatique : note sectorielle*, Eurostaf, 91 p.
- EUROSTAF (1997 B), *Les SSII indépendantes*, Eurostaf.

- EYMARD-DUVERNAY François (1994), « Coordination des échanges par l'entreprise et qualité des biens », pp. 307-334, in ORLEAN André (éd.), *Analyse économique des conventions*, PUF (Economie), 403 p.
- FAGUER Jean-Pierre, GOLLAC Michel (1997), « Ordinateur universel ou personnel ? Clarté et ambiguïté dans la définition des techniques », pp. 97-122, in CONEIN Bernard, THEVENOT Laurent (eds.), *Cognition et information en société*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales, (Raisons pratiques 8), 317 p.
- FARELL J., SALONER G. (1986), « Installed base and compatibility : Innovation, products pre-announcements and predation », in *American Economic Review*, vol. 16, 5, pp. 940-955.
- FAVEREAU Olivier (1986), « La formalisation du rôle des conventions dans l'allocation des ressources », pp. 249-267, in SALAIS Robert, THEVENOT Laurent (eds.), *Le travail : marchés, règles, conventions*, Economica / INSEE, 370 p.
- FAVEREAU Olivier (1989), « Marchés internes, marchés externes », in *Revue Economique* n° 3, pp. 273-328.
- FAVEREAU Olivier (1994), « Règle, organisation et apprentissage collectif : un paradigme non standard pour trois théories hétérodoxes », pp. 113-137, in ORLEAN André (éd.), *Analyse économique des conventions*, PUF (Economie), 403 p.
- FAVEREAU Olivier (1998), « Notes sur la théorie de l'information à laquelle pourrait conduire l'économie des conventions », pp.195-238, in PETIT Pascal (éd.), *L'économie de l'information : les enseignements des théories économiques*, La Découverte (Recherches), 405 p.
- FILIPPI Maryline, PIERRE Emmanuel, TORRE André (1996), « Quelles approches économiques pour la notion de réseau ? : Contenus théoriques et dimensions opérationnelles », in *Revue d'Economie Industrielle* n° 77, pp. 87-98.
- FLETCHER Patricia D., DIAMOND Lester (1997), « L'organisation basée sur l'information : gérer la productivité de la force de travail », pp. 167-181, in MONNOYER-LONGE Marie-Christine (éd.), *L'entreprise et l'outil informationnel*, L'Harmattan, 220 p.
- FORAY Dominique (1989), « Les modèles de compétition technologique : Une revue de la littérature », in *Revue d'Economie Industrielle*, n° 48, pp. 16-34.
- FORAY Dominique (1990), « Exploitation des externalités de réseau versus évolution des normes : les formes d'organisation face au dilemme de l'efficacité dans le domaine des technologies de réseau », in *Revue d'Economie Industrielle* n° 51, pp. 113-140.
- FORAY Dominique (1995), « Les brevets dans la nouvelle économie de l'innovation », pp. 119-149, in BASLE Maurice (et al.), *Changement institutionnel et changement technologique : évaluation, droits de propriété intellectuelle, système national d'innovation*, CNRS Editions, 333 p.
- FORAY Dominique (1996), « Diversité, sélection et standardisation : les nouveaux modes de gestion du changement technique », in *Revue d'Economie Industrielle* n° 75, pp. 257-274.
- FORAY Dominique et FREEMAN Christopher (éditeurs) (1992), *Technologie et richesse des nations. Colloque Technologie et compétitivité*, juin 1990, Paris, Economica, 517 p.
- FORAY Dominique, LUNDEVALL Bengt Ake (Eds.) (1996), *Employment and Growth in the Knowledge-Based Economy*, OCDE, 389 p.
- FORAY Dominique, LUNDEVALL Bengt Ake (1997), « Une introduction à l'économie fondée sur la connaissance », pp. 16-38, in GUILHON Bernard, HUARD Pierre, ORILLARD Magali, ZIMMERMANN Jean-Benoît (eds) (1997), *Economie de la connaissance et organisations : Entreprises, Territoires, Réseaux*, L'Harmattan, 481 p.
- FRANKE R. H. (1987), « Technical revolution and productivity decline : computer introduction in the financial industry », in *Technical Forecast Society Change*, April, Vol.31, n° 2, pp. 143-154.
- FREEMAN Chris (1988), « Diffusion : la propagation des nouvelles technologies dans les entreprises, les différents secteurs et Etats », pp. 34-63, in HEERTJE Arnold (éd.), *Innovation, technologie et finance*, Basil Blackwell, 199 p.

- FREEMAN Chris (1995), « Le nouveau contexte de l'innovation », in *STI Revue* n°15, reproduit dans *Problèmes économiques* n° 2464-2465, sous le titre « Les pouvoirs publics entre néo-keynésianisme et néo-schumpéterisme », pp. 28-35.
- FREIDSON Eliot (1984), *La profession médicale*, Payot (Médecine et sociétés), traduction de *Profession of Medicine*, Harper & Row (1970), 369 p.
- FREYSSINET M. (1995), « La production réflexive : une alternative à la "production de masse" et à la "production au plus juste" », in *Sociologie du travail*, n° 3.
- FRIEDMANN Georges, NAVILLE Pierre (eds.) (1962), *Traité de sociologie du travail*, Armand Colin, 468 p.
- GABRIE Hubert, JACQUIER Jean-Louis (1994), *La théorie moderne de l'entreprise : l'approche institutionnelle*, Economica, 330 p.
- GADREY Jean (1987), « Productivité et évaluation des services : la construction sociale du produit », in *Cahiers ERMES* n° 2, Janvier 1987.
- GADREY Jean (1988), « Des facteurs de croissance des services aux rapports sociaux de services », in *Revue d'Economie Industrielle* n° 43, pp. 34-48.
- GADREY Jean (1991), « Le service n'est pas un produit : quelques implications pour l'analyse économique et pour la gestion », in *Politiques et management public*, vol. 9, n° 1, mars, pp. 1-24.
- GADREY Jean (1994 A), « La modernisation des services professionnels : Rationalisation industrielle ou rationalisation professionnelle ? », in *Revue Française de Sociologie*, XXXV, avril-juin 1994, pp. 163-195.
- GADREY Jean (1994 B), « Relations, contrats et conventions de service », pp. 123-153., in DE BANDT Jacques, GADREY Jean (eds.), *Relations de service, marchés de service*, CNRS Editions (Recherches et entreprise), 360 p.
- GADREY Jean (1994 C), « Les relations de service dans le secteur marchand », pp. 23-43, in DE BANDT Jacques, GADREY Jean (eds.), *Relations de service, marchés de service*, CNRS Editions (Recherches et entreprise), 360 p.
- GADREY Jean (1996 A), *Services, la productivité en question*, Desclée de Brouwer, 359 p.
- GADREY Jean (1996 B), *Note sur les mondes de production*, Ronéo, CLERSE.
- GADREY Jean (1999 A), « La caractérisation des biens et des services, d'Adam Smith à Peter Hill : une approche alternative », ronéo, 30 p.
- GADREY Jean (1999 B), « Flexibilité et professionnalisation du travail dans les services : des stratégies et des modèles distincts », in VI^{èmes} journées IFRESI, volume 2, *Organisation du travail, Gestion de l'emploi, des compétences et de la formation*, pp. 149-164.
- GADREY Jean (1999 C), *Vingt-cinq ans de productivité du travail à La Poste*, ronéo, 42 p.
- GADREY Jean (1999 D), « Critique du paradigme industrialiste » pp. 5-19, in ASSOCIATION NATIONALE DE LA RECHERCHE TECHNIQUE, *L'innovation dans les services : une invitation à l'insurrection intellectuelle*, Economica, 98 p.
- GADREY Jean, DELAUNAY Jean-Claude (1987), *Les enjeux de la société de service*, Presses de la Fondation Nationale des Sciences Politiques, 333 p.
- GADREY Jean, NOYELLE Thierry, STANBACK Thomas, *Les rendements décroissants du concept de productivité dans l'analyse de l'efficacité du travail et des performances économiques*, Ronéo.
- GAFFARD Jean-Luc (1990), *Economie industrielle et de l'innovation*, Dalloz (Précis), 470 p.
- GALLOUJ Faïz (1994) *Economie de l'innovation dans les services*, L'Harmattan, 255 p.
- GALLOUJ Faïz (1997), *Vers une théorie de l'innovation dans les services*, ronéo, CLERSE, 149 p.
- GALLOUJ Faïz, WEINSTEIN Olivier (1997), « Innovation in services », in *Research Policy*, n° 26, pp. 537-556.
- GARROUSTE Pierre, GONZALEZ Sylvie (1995), « Information, processus d'apprentissage, et évolution d'un réseau : le cas du GBM », in *Economies et Sociétés, Série Dynamique technologique et organisation*, W, n°2, pp. 119-138.
- GEIB Jean-Marc (1989), *Cours de génie logiciel*, LIFL Université des Sciences et des Techniques de Lille, 115 p.

- GENTHON Christian (1995), *Croissance et crise de l'industrie informatique mondiale*, Syros (Alternatives économiques), 189 p.
- GENTHON Christian, PHAN Denis (1999), « Les logiciels libres : un nouveau modèle ? », in revue *Terminal, Technologie de l'Information, Culture et Société*, n° 80/81, « Les logiciels libres : de l'utopie au marché », pp. 167-188.
- GERARD VARET L.A., ZIMMERMANN J.B. (1980), *Le concept de produit informatique, Première partie : un modèle général*, rapport BNI n° 30.
- GERSHUNY Jonathan (1978), *After Industrial Society ? : The emerging Self-Service Economy*, Mac Millan.
- GIANAZZA David (et al.) (1997), « Validation d'un système critique par rétro-conception formelle », *Génie Logiciel* n° 46, pp. 118-122.
- GIARINI Orio, STAHEL Walter R. (1990), *Les limites du certain : affronter les risques dans une nouvelle économie de service*, Presses polytechniques et universitaires romandes (Série d'informations du club de Rome), 191 p.
- GIBBS M. (1994), *The Crisis in Software*, Scientific American.
- GIRIN Jacques (1994) *Les agencements organisationnels, Actes du séminaire « Contradictions et Dynamique des Organisations*», Vii-Frog, 32 p.
- GOLDFINGER Charles (1994), *L'utile et le futile : l'économie de l'immatériel*, Editions Odile Jacob, 622 p.
- GOMEZ Pierre-Yves (1994), *Qualité et théorie des conventions*, Economica (Recherche en gestion), 270 p.
- GOMEZ Pierre-Yves (1997), « Information et conventions : le cadre du modèle général », in *Revue Française de Gestion* n° 112, pp. 64-77.
- GRANOVETTER Mark (1985), « Economic Action and Social Structure : the Problem of Embeddedness », in *American Journal of Sociology*, vol. 91, n° 3, pp. 481-510.
- GRANOVETTER Mark (1990), « The Old and the New Economic Sociology », in FRIEDLAND R. et ROBERTSON A.F. (eds.), *Beyond the marketplace : rethinking economy and society*, A. de Gruyter, (Sociology and economics, Controversy and integration), 365 p.
- GRANOVETTER Mark (1994), « La construction économique des institutions », in ORLEAN André (éd.) *Analyse économique des conventions*, PUF (Economie), 403 p.
- GRAY Pamela (1993), *Les systèmes ouverts : une stratégie d'entreprise pour les années 90*, Ediscience, 332 p.
- GROSSMAN G.M., HELPMAN E. (1994), « Endogenous Innovation in the Theory of Growth », in *Journal of Economic Perspectives*, vol. 8, n° 1.
- GUILHON Bernard (1991), « Rendements et productivité », pp. 524-538, in ARENA Richard et ali (eds), *Traité d'économie industrielle*, Economica, 1001 p.
- GUILHON Bernard, HUARD Pierre, ORILLARD Magali, ZIMMERMANN Jean-Benoît (eds) (1997), *Economie de la connaissance et organisations : Entreprises, Territoires, Réseaux*, L'Harmattan, 481 p.
- GUIOT Jean M., BEAUFILS Alain (eds.) (1988), *Design de l'organisation : du poste de travail à la mega-structure*, Gaëtan Morin Editeur, 202 p.
- HABERMAS Jürgen (1987), *Théorie de l'agir communicationnel*, Fayard (L'espace du politique).
- HAINAUT Jean-Luc (1994), *Bases de données et modèles de calcul : Outils et méthodes pour l'utilisateur*, InterEditions, 357 p.
- HALSTEAD Maurice H. (1977), *Elements of software science*, Elsevier, 127 p.
- HAMON Jean-Paul (1996), « La conduite du développement d'un grand système d'informations : l'expérience du projet SIMAT », in *Génie Logiciel* n° 40, pp. 15-19.
- HARVEY Mara C. (1998), « Ecart de productivité et « maladie des coûts : Apports et limites du modèle de croissance déséquilibrée de William J. Baumol », in *Revue Economique*, vol. 49, n° 2, pp. 437-467.
- HATCHUEL Armand (1996), *Intervention au séminaire « Economie de l'information »*, Commissariat Général du Plan, ronéo.
- HATCHUEL Armand, WEIL Benoît (1992), *L'expert et le système : gestion des savoirs et métamorphose des acteurs dans l'entreprise industrielle, suivi de Quatre histoires de systèmes-experts*, Economica, 263 p.

- HAWKINS Richard (1995), « Le secteur public et la normalisation des technologies de l'information », in *STI Revue* n° 16, OCDE, pp. 167-195.
- HEERTJE Arnold (éd.) (1988), *Innovation, technologie et finance*, Basil Blackwell, 199 p
- HEERTJE Arnold (1988), « Innovation technologique et financière », pp. 1-12, in HEERTJE Arnold (éd.).
- HERAUD Jean-Alain (1995), « Brevets et contexte institutionnel de la création technologique », pp. 91-117, in BASLE Maurice (et al.), *Changement institutionnel et changement technologique : évaluation, droits de propriété intellectuelle, système national d'innovation*, CNRS Editions (Recherche et entreprise), 333 p.
- HILL Peter (1997), « Tangibles, Intangibles and services : a New Taxonomy for the Classification of Output », in *CSLS Conference on Service Productivity*, 22 p.
- HIRSCHMAN Albert O. (1986), *Vers une économie politique élargie*, Editions de Minuit (Le sens commun), 111 p.
- HIRSCHMAN Albert O. (1972), *Face au déclin des entreprises et des institutions*, Editions Ouvrières (Collection Economie humaine), 141 p., traduction de *Exit, Voice and loyalty : responses to decline in firms, organizations and states*, (1970), Harvard University Press.
- HITT L.M., BRYNJOLFSSON E. (1996), « Productivity, Business Profitability, and Consumer Surplus : Three Different Measures of Information Technology value », in *MIS Quarterly*, June, Vol 20, n°2, pp. 121-142.
- HORN François (1998), « Processus de standardisation et diversité de l'information fournie à l'aide de moyens informatiques », pp. 135-150, in ALLAIN Annie, ESSAMA Gervais (eds), *Libre échange et identité culturelle*, Presses Universitaires du Septentrion (UL3), 282 p.
- HORN François (1999 A), « Diversité des informations traitées par des moyens informatiques, standardisation optimale et acteurs du processus de standardisation », in *Communications et stratégies* n°33, pp. 85-117.
- HORN François (1999 B), « L'importance du logiciel libre dans l'amélioration de l'efficacité des logiciels et de leur processus de production », in *revue Terminal, Technologie de l'Information, Culture et Société*, n° 80/81, « Les logiciels libres : de l'utopie au marché », pp. 117-148.
- HORN François (1999 C), « L'analyse de l'évolution de l'informatique : la constitution successive de réseaux technico-économiques structurés par des standards », in *La conduite des activités économiques : logiques managériales et logiques sectorielles, VI^{èmes} journées IFRESI*, volume 1, pp. 35-48.
- ICHBIAH Daniel (1995), *Bill Gates et la saga de Microsoft*, Pocket, 392 p.
- IGAEN (1998), *Observations sur le fonctionnement des établissements d'enseignement supérieur*, Rapport de l'Inspection Générale de l'Administration de l'Education Nationale, 54 p.
- ILLICH Ivan (1973), *La convivialité*, Seuil, 157 p., Trad. de *Tools for conviviality*.
- JACOBSON Ivar (1993), *Le génie logiciel orienté objet : une approche fondée sur les cas d'utilisation*, Addison-Wesley, 536 p.
- JAULENT Patrick (1992) *Génie logiciel : les méthodes : SADT, SA, E-A, SA-RT*, Armand Colin, 259 p.
- JOLY Pierre-Benoît, MANGEMATIN Vincent (1995), « Les acteurs sont-ils solubles dans les réseaux ? », in *Economies et Sociétés, Série Dynamique technologique et organisation*, W, n°2, pp. 17-50.
- JONES Capers (1989), *La productivité en génie logiciel*, Les Editions d'Organisation, (Collection ingénierie des systèmes d'information), 329 p., traduction de *Programming Productivity*, Mac Graw Hill. Inc (1986).
- JONES Capers (1998), « Etre cacique en logiciel : servitudes et grandeurs », in *Génie logiciel*, n° 47, mars 1998, pp. 19-38.
- JONSCHER C. (1984), « Economie et politique de l'information », in *Le bulletin de l'IDATE* n° 16.
- JULLIEN Nicolas (1999), « Linux : la convergence du monde Unix et du monde PC », in *revue Terminal, Technologie de l'Information, Culture et Société*, n° 80/81, « Les logiciels libres : de l'utopie au marché », pp. 41-70.
- KAHIN B. (1990), « The Software patent crisis », in *Technology Review*, avril.
- KANDIL Feriel (1998), « De la rationalité à la raison pratique dans les actes économiques », pp. 145-172, in SALAIS Robert (éd.) (1998 A), *Institutions et convention : La réflexivité de l'action économique*, EHESS, (Raisons pratiques 9), 304 p.

- KAPLAN Daniel (1999), L'emploi, le travail et l'entreprise, modelés par les NTIC, <http://www.dkaplan.net/emploi.htm>
- KARPIK Lucien (1989), « L'économie de la qualité » in *Revue Française de Sociologie* n°XXX, pp. 182-210.
- KATZ M. et SHAPIRO C. (1985), « Network externalities, competition and compatibility », in *American Economic Review*, 75, 3, pp. 424-440.
- KIVIJARVI H., SAARINEN T. (1995), « Investment in information systems and the financial performance of the firm », in *Information and Management* n° 28, pp. 143-163.
- KNIGHT Frank Hyneman (1964), *Risk, Uncertainty and Profit*, Houghton Mifflin [1921], 381 p.
- KNORR CETINA Karin (1997), « Les métaphores dans les laboratoires scientifiques », pp. 47-64 in CONEIN Bernard, THEVENOT Laurent (eds.), *Cognition et information en société*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Raisons pratiques 8), 317 p.
- KUHLEN Rainer (1997), « Les effets de valeur ajoutée des marchés de l'information », pp. 171-180, in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- KUHN T.H. (1970), *The structure of scientific revolution*, Chicago University Press, (traduction française, *la structure des révolutions scientifiques*, Flammarion, 1972), 210 p.
- LABAT Eric (1984) *L'industrie française du logiciel et des services informatiques et la concurrence internationale* Thèse de Troisième Cycle, Paris 1.
- LAMBERTON Donald (1997), « Pour une taxonomie de l'information », pp. 69-85, in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- LAMIZET Bernard, SILEM Ahmed (Eds.) (1997), *Dictionnaire encyclopédique des sciences de l'information et de la communication*, Ellipses, 590 p.
- LANCASTER K.J. (1966), « A New Approach to Consumer Theory », in *Journal of Political Economy*, 14, pp. 133-156.
- LANDAUER Thomas K. (1997), *The Trouble with Computers : Usefulness, Usability, and Productivity*, MIT Press, 425 p.
- LANG Bernard (1997), « Ressources libres et indépendance technologique dans les secteurs de l'information », *Colloque Inforoutes et technologies de l'information*, Forum Initiatives 97, 10 p., <http://pauillac.inria.fr/~lang/ecrits/hanoi/>, article reproduit dans *Technique et science informatique*, vol. 18, n° 8, oct. 1999, pp. 901-914.
- LANG Bernard (1998 A), « Les logiciels libres à la disposition de tous » in *Le Monde Diplomatique*, Janvier 1998.
- LANG Bernard (1998 B), « Oui au libéralisme technologique », in *01 Informatique*, n° 1494, p. 98.
- LANG Bernard (1998 C), « Ce modèle économique est pérenne », in *Le Monde Informatique*, n° 774, p. 98.
- LANGLOIS Richard N., MOWERY David C. (1996), « The Federal Government Role in the Development of the U.S. Software Industry », pp. 53-86, in MOWERY David C. (ed.), *The international computer software industry : a comparative study of industry evolution and structure*, Oxford University Press, 324 p.
- LATOUR Bruno (1993), « Le topofil de Boa-Vista. La référence scientifique : montage photo-philosophique », pp. 187-216, in CONEIN Bernard, DODIER Nicolas et THEVENOT Laurent (eds.), *Les objets dans l'action : De la maison au laboratoire*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Raisons pratiques 4), 290 p.
- LAWRENCE David B. (1997), *Evaluation et acquisition d'une source d'information*, pp. 197-213, in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- LAWRENCE Paul R. et LORSCH Jay W. (1989), *Adapter les structures de l'entreprise*, Editions d'Organisation, (Les Classiques E.O), 237 p., traduction de *Organization and Environment*, Harvard School Business Press (1986).
- LE DORTZ Laurent (1997), « Le rôle des connaissances dans la dynamique industrielle : irréversibilités et émergence des industries », pp. 232-262, in GUILHON Bernard, HUARD Pierre, ORILLARD Magali, ZIMMERMANN Jean-Benoît (eds) (1997), *Economie de la connaissance et organisations : Entreprises, Territoires, Réseaux*, L'Harmattan, 481 p.

- LE MOIGNE Jean-Louis (1977), *La théorie du système général : théorie de la modélisation*, PUF (Systèmes-décisions), 258 p.
- LE MOIGNE Jean-Louis (1990), *La modélisation des systèmes complexes*, Dunod (Afcet systèmes), 178 p.
- LE MOIGNE Jean-Louis (1998), « La modélisation systémique de l'information », pp.55-73, in PETIT Pascal (éd.) (1998), *L'économie de l'information : les enseignements des théories économiques*, La Découverte, (Recherches), 405 p.
- LE NAGARD Emmanuelle (1997), « Les stratégies de compatibilité dans les industries de la communication », in *Communications et Stratégies* n° 27, pp. 103-129.
- LEQUILLER François (1997), « L'indice des prix à la consommation surestime-t-il l'inflation ? », in *Economie et Statistiques*, n° 303, pp. 3-33.
- LEVY Pierre (1992), *De la programmation considérée comme un des beaux-arts*, La Découverte, 245 p.
- LEWIS David K. (1969), *Convention : a philosophical study*, Harvard University Press, 213 p.
- LIAGOURAS Georges (2000), « Evolutionnisme et économie des conventions : Sur les limites symétriques de deux hétérodoxies actuelles », in *Economies et Sociétés*, Série « Théorie de la régulation », R, n° 11, 1/2000, pp. 183-205.
- LIVET Pierre et THEVENOT Laurent (1994), « Les catégories de l'action collective », pp. 139-168, in ORLEAN André (éd.) (1994), *Analyse économique des conventions*, PUF (Economie) 403 p.
- LUCAS André (1987), *Le droit de l'informatique*, PUF (Thémis Droit), 551 p.
- LUCAS Yvette (1989), *Le Vol du savoir : techniciens de l'aéronautique et évolution des technologies*, Presses Universitaires de Lille (Sociologie, Mutations), 255 p.
- LUNDEVALL Bengt-Ake (ed.) (1992), *National systems of innovation : towards a theory of innovation and interactive learning*, Pinter, 342 p.
- LUNDEVALL Bengt-Ake (1997), « Information Technology in the Learning Economy : challenges for development strategies », in *Communications et Stratégies*, n° 28, pp. 177-192.
- MACCABE Thomas J. (1976), « A Complexity Measure », in *IEEE Transactions on Software Engineering*, Vol. SE-2, pp. 308-320.
- MACHLUP Fritz (1962), *The production and distribution of knowledge in the United States*, Princeton University Press, 416 p.
- MACHLUP Fritz (1984), *The economics of information and human capital*, Princeton University Press (Knowledge), 644 p.
- MARCH James Gardner (1991), *Décisions et organisations*, Ed d'Organisation (Les Classiques E.O.), 275 p.
- MARIOTTI Sergio (1997), « Technologies de l'information et de la communication : conséquences pour l'emploi » in *Communications et Stratégies* n° 28, pp. 149-176.
- MARX Karl (1857), « Fragment sur les machines », in *Œuvres*, vol. II, Bibliothèque de la Pléiade, 1968, pp. 304-316.
- MASSARD Nadine (1997), « Externalités, apprentissage et dynamique technologique. Un retour sur la notion de réseau », pp. 336-355, in GUILHON Bernard, HUARD Pierre, ORILLARD Magali, ZIMMERMANN Jean-Benoît (eds.), *Economie de la connaissance et organisations : Entreprises, Territoires, Réseaux*, L'Harmattan, 488 p.
- MAURICE Marc, SELLIER François, SILVESTRE Jean-Jacques (1982), *Politique d'éducation en France et en Allemagne*, PUF (Sociologies), 382 p.
- MAYER René (1997), « L'information en questions », pp. 61-66, in MAYERE Anne (éd.), *La société informationnelle : Enjeux sociaux et approches économiques*, L'Harmattan (Communication et civilisation), 239 p.
- MAYERE Anne (1990), *Pour une économie de l'information*, Editions du CNRS, 316 p.

- MAYERE Anne (1994), « Relation de service et enjeux d'industrialisation », in DE BANDT Jacques, GADREY Jean, *Relations de service, marchés de services*, CNRS Editions (Recherche et entreprise), 357 p.
- MAYERE Anne (éd.) (1997), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- MAYERE Anne (1997 A) « Produits et services d'information : proposition de dépassement de la théorie standard de l'information », pp. 125-147, in MAYERE Anne (éd.)
- MAYERE Anne (1997 B), « Capitalisation des connaissances et nouveau modèle industriel », pp. 197-213, in MONNOYER-LONGE Marie-Christine (éd.), *L'entreprise et l'outil informationnel*, L'Harmattan, 220 p.
- MAYNARD-SMITH J. et PRICE G.R. (1973), « The Logic of Animal Conflict », in *Nature*, n° 246, pp. 15-18.
- MENARD Claude (1990), *L'économie des organisations*, La Découverte (Repères), 127 p.
- MERGES Robert (1996), « A Comparative Look at Property Rights and the Software Industry », pp. 272-303, in MOWERY David C. (ed.), *The international computer software industry : a comparative study of industry evolution and structure*, Oxford University Press, 324 p.
- MICHON Robert, GINGRAS Lin (1988), « Le processus d'implantation des systèmes d'information : l'approche traditionnelle et les approches itératives », pp. 121-134, in GUIOT Jean M., BEAUFILS Alain (eds.), *Design de l'organisation : du poste de travail à la mega-structure*, Gaëtan Morin Editeur, 202 p.
- MILON Alain (1999), *La valeur de l'information entre dette et don : critique de l'économie de l'information*, PUF (Sociologie d'aujourd'hui), 232 p.
- MINASI Mark (2000), *The software conspiracy : why software companies put out faulty products, how they can hurt you, and what you can do about it*, McGraw Hill, 269 p.
- MINTZBERG Henry (1982), *Structure et dynamique des organisations*, Ed. d'Organisation, 434 p.
- MINTZBERG Henry (1990), *Le management : voyage au centre des organisations*, Ed. d'Organisation, 570 p.
- MIYAKAWA Takayasu (1997), « L'information dans l'industrie, une analyse des expériences japonaises », pp. 67-82, in MONNOYER-LONGE Marie-Christine (éd.), *L'entreprise et l'outil informationnel*, L'Harmattan, 220 p.
- MONNOYER-LONGE Marie-Christine (éd.) (1997), *L'entreprise et l'outil informationnel*, L'Harmattan, 220 p.
- MONNOYER-LONGE Marie-Christine (1997), Système d'information et productivité : investigation d'un paradoxe, pp. 107-123, in MONNOYER-LONGE Marie-Christine (éd.)
- MOSHOWITZ Abbe (1997), « Le meilleur des marchés : marchandises informationnelles et changement de société », pp. 23-42, in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- MOULINE Abdelaziz (1996), *L'Industrie des services informatiques*, Economica, (Economie des services poche), 109 p.
- MOWERY David C. (ed.) (1996), *The international computer software industry : a comparative study of industry evolution and structure*, Oxford University Press, 324 p.
- NEGROPONTE Nicholas (1995), *L'homme numérique*, Robert Laffont, (Pocket) 295 p.
- NELSON P. (1970), « Information and consumer behavior », in *Journal of Political Economy*, n° 78, pp. 311-329.
- NELSON R., WINTER S. (1975), « Growth Theory from an Evolutionary Perspective : The Differential Productivity Puzzle », in *American Economic Review*, vol. 65, n° 2, pp. 338-344.
- NIOSI Jorge (2000), *Alliances, masse critique et exportations : la croissance rapide dans l'industrie québécoise du logiciel*, Communication à la conférence internationale, Economie et socio-économie des services : perspectives internationales, Lille, 22 et 23 juin 2000.
- NIOSI Jorge, BELLON Bertrand, SAVIOTTI Paolo, CROW Michaël (1992), « Les systèmes nationaux d'innovation : à la recherche d'un concept utilisable » in *Revue Française d'Economie*, n° 7, pp. 215-249.
- NONAKA Ikujiro, TAKEUCHI Hirotaka (1997), *La connaissance créatrice : La dynamique de l'entreprise apprenante*, De Boeck Université (Management), 303 p.
- NORMAN D.A. (1993), « Les artefacts cognitifs », pp. 15-34, in CONEIN Bernard, DODIER Nicolas et THEVENOT Laurent ; *Les objets dans l'action. De la maison au laboratoire*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Raisons pratiques 4), 290 p.

- OCDE (1991 A) *Le génie logiciel : un défi pour l'action gouvernementale*, OCDE (PIIC Politiques d'Information, d'Informatique et de Communications n°26), 66 p.
- OCDE (1991 B), *La dimension économique des normes en matière de technologies de l'information*, OCDE (PIIC Politiques d'Information, d'Informatique et de Communications n°25), 116 p.
- OCDE (1991 C), *Technology and productivity : The Challenge for Economic Policy*, OCDE, 588 p.
- OCDE (1992), *Technology and the Economy : The Key Relationships*, OCDE (TEP Report), 328 p.
- OCDE (1993 A), *La mesure des activités scientifiques et technologiques 1993 : Méthode type proposée pour les enquêtes sur la recherche et le développement expérimental, Manuel de Frascati*, OCDE, 284 p.
- OCDE (1993 B), *Les indicateurs d'utilisation : Une base nouvelle pour les politiques des technologies de l'information*, OCDE (PIIC – Politiques d'information, d'informatique et de communications), 129 p.
- OCDE (1994), *Perspectives des technologies de l'information : 1994*, OCDE (PIIC – Politiques d'information, d'informatique et de communications), 88 p.
- OCDE (1995 A), *Vers la réalisation de la société de l'information*, Résumé final du Président, Session spéciale du Comité de la politique de l'information, de l'informatique et des communications sur les infrastructures de l'information.
- OCDE (1995 B), *Perspectives des technologies de l'information : 1995*, 156 p.
- OCDE (1996), *L'économie fondée sur le savoir*, OCDE, 47 p.
- OCDE (1997, A), *L'infrastructure mondiale de l'information et la société mondiale de l'information (GII-GIS) : les politiques requises*, 84 p.
- OCDE (1997, B) *Perspectives des technologies de l'information : 1997*, OCDE, 262 p.
- OCDE (1998 A), *Le secteur du logiciel : un profil statistique pour certains pays de l'OCDE*, OCDE, 48 p.
- OCDE (1998 B), *Les contenus : un nouveau secteur de croissance*, OCDE, 27 p.
- OFFICE OF TECHNOLOGY ASSESSMENT (1992), *Finding a Balance, Computer Software, intellectual property and the challenge for technological change*, Congress of the United States.
- ORLEAN André (éd.) (1994), *Analyse économique des conventions*, PUF (Economie), 403 p.
- ORLEAN André (1996), *La théorie économique de la confiance et ses limites*, communication au colloque de l'Association pour le Développement de la Socio-Economie, 20 p.
- ORLEAN André et TADJEDDINE Yasmina (1998), « Efficacité informationnelle et marchés financiers », pp.153-195, in PETIT Pascal (éd.) *L'économie de l'information : les enseignements des théories économiques*, La Découverte (Recherches), 405 p.
- PALIERNE Catherine (1998), *L'avantage informatique : 50 dirigeants témoignent*, Ed. d'Organisation, 206 p.
- PALLOIX Christian, RIZOPOULOS Yorgos (eds.) (1997), *Firmes et économie industrielle*, L'Harmattan, (Economie et innovation), 447 p.
- PASSERON Jean-Claude (1991), *Le raisonnement sociologique : l'espace non-poppérien du raisonnement naturel*, Nathan (Essais et recherches), 408 p.
- PAULRE Bernard (1996), *L'Economie de l'information : Quelques éléments de réflexion complémentaires*, ronéo, 19 p.
- PAVE Francis (1989), *L'illusion informaticienne*, L'Harmattan (Logiques sociales), 270 p.
- PAVITT K. (1984), « Sectoral Patterns of Technical Change : Towards a Taxonomy and a Theory », in *Research Policy*, 13, pp. 343-373.
- PEAUCELLE Jean-Louis (1981), *Les systèmes d'information : la représentation*, PUF (Systèmes-décision), 249 p.
- PEAUCELLE Jean-Louis (1997), *Informatique rentable et mesure des gains*, Hermès (Informatique et gestion), 238 p.
- PEAUCELLE Jean-Louis (1998), *La productivité administrative et l'informatique : discours et réalités*, contribution au 11^{ème} Colloque Européen en Informatique et Société, Strasbourg, 10-12 juin 1998, 9 p.

- PEREZ C. (1983), « The Assimilation of New Technologies in the Social and Economic System », in *Futures*, vol. 13, n° 5, pp. 357-375.
- PERROT Anne (1995), « Ouverture à la concurrence dans les réseaux », in *Economie et prévision*, reproduit sous le titre « L'économie des réseaux », in *Problèmes Economiques* n°2456, pp. 7-14.
- PETIT Pascal (1994), « Formes de services et modes d'internationalisation des économies », pp. 281-306, in DE BANDT Jacques, GADREY Jean (eds.), *Relations de service, marchés de service*, CNRS Editions (Recherches et entreprise), 360 p.
- PETIT Pascal (éd.) (1998), *L'économie de l'information : les enseignements des théories économiques*, La Découverte (Recherches), 405 p.
- PETIT Pascal (1998 A), « L'économie de l'information au crible des théories de l'information », pp. 341-396, in PETIT Pascal (éd.).
- PETIT Pascal (1998 B), « L'économie de l'information en questions », pp. 15-35, in PETIT Pascal (éd.).
- PETIT Pascal (1999), *Les aléas de la croissance dans une économie fondée sur le savoir*, document CEPREMAP n° 99, 35 p.
- PICAVET Monique (1997), *La complexité dans la modélisation du système d'information de l'entreprise. Propositions de solutions : concepts, outils et démarches*, HDR, Université Lille 1, 200 p.
- PICHAULT François (1990), *Le conflit informatique : gérer les ressources humaines dans le changement technologique*, Ed. Universitaire / De Boeck Université (Collection Ouvertures sociologiques), 259 p.
- POLANYI Karl (1983), *La Grande Transformation : aux origines politiques et économiques de notre temps*, Gallimard (Collection Bibliothèque des Sciences Humaines), 419 p.
- POLANYI Michael (1966), *The Tacit Dimension*, Routledge and Kegan Paul, 108 p.
- PORAT Marc Uri (1997), *The Information Economy*, special publication OT 77-12, 9 vols., US Department of Commerce.
- PORTER Michael E. (1982), *Choix stratégiques et concurrence : techniques d'analyse des secteurs et de la concurrence dans l'industrie*, Economica, 426 p.
- POSTEL Nicolas (2000), *Règles et rationalité économique*, Thèse de doctorat en Sciences Economiques, Université de Lille I, 386 p.
- PRAKKE Frits (1988), « Le financement de l'innovation technologique », pp. 64-91, in HEERTJE Arnold (éd.), *Innovation, technologie et finance*, B. Blackwell, 199 p.
- PRINTZ Jacques (1998), *Puissance et limites des systèmes informatisés*, Hermès (Etudes et logiciels informatiques), 351 p.
- QUINN James Brian (1992), *Intelligent Enterprise : A Knowledge and Service Based Paradigm for Industry*, Free Press / Maxwell McMillan, 473 p.
- RALLET Alain (1997), « L'efficacité des technologies de l'information et de la communication à l'étape des réseaux », pp. 85-106, in MONNOYER-LONGE Marie-Christine (éd.), *L'entreprise et l'outil informationnel*, L'Harmattan, 220 p.
- RAMAUX Christophe (1996), *Marché, contrats, règles et institutions : les spécificités de la relation salariale*, Thèse de doctorat en Sciences Economiques, Université de Paris I, 388 p.
- RAYMOND Eric S. (1998), *La cathédrale et le bazar*, traduction de BLONDEEL Sébastien, http://www.lifl.fr/~blondeel/traduc/Cathedral-bazaar/Main_file.html
- RAYMOND Eric S. (2000), « A la conquête de la noosphère », in BLONDEAU Olivier, LATRIVE Florent (2000), *Libres enfants du savoir numérique*, L'Eclat, 502 p.
- REYNAUD Jean-Daniel (1989), *Les règles du jeu : l'action collective et la régulation sociale*, Armand Colin (collection U), 306 p.
- RIBAULT Thierry (1993), *Economie de l'information : approche patrimoniale*, A Jour (Nouvelles technologies documentaires), 355 p.

- RIBAULT Thierry (1995), « Gestion patrimoniale de l'information et capitalisme relationnel : le cas du Japon », in *Les performances organisationnelles*, IV^{èmes} journées IFRESI, CNRS, p.216-228.
- RIVAUD-DANSET Dorothée (1998), « Le traitement de l'incertitude en situation : une lecture de Knight », pp. 23-50, in SALAIS Robert (éd.), *Institutions et conventions : la réflexivité de l'action économique*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Raisons pratiques 9), 303 p.
- RIVIERE Philippe (1999), « Le piège des brevets informatiques », in *Le Monde Diplomatique*, Mars.
- ROACH S.S. (1989), « America's white collar productivity dilemma », in *Manufacturing Engineering*, August, 104 p.
- ROBERT Philippe (1997), « Définir les exigences qualité pour limiter les risques lors des développements de logiciels », in *Génie Logiciel*, n° 46, pp. 167-172.
- ROMER Paul (1994), « The Origins of Endogenous Growth », in *The Journal of Economic Perspectives*, vol. 8.
- ROSENBERG Nathan. (1983), *Inside the black box : technology and economics*, Cambridge University Press, 304 p.
- ROUX Frédéric Georges (1991), *Infocentre : Pourquoi ? Comment ?*, Eyrolles, 297 p.
- ROWE Frantz (1994), *Des Banques et des Réseaux : Productivité et avantages concurrentiels*, ENSPTT Economica (Management – Communication – Réseaux), 360 p.
- ROWE Frantz (1997), « Productivité de l'information et design organisationnel, accessibilité aux données et agir communicationnel », pp. 23-40, in MONNOYER-LONGE Marie-Christine (éd.), *L'entreprise et l'outil informationnel*, L'Harmattan, 220 p.
- RUBIELLO Luc (1997), *Techniques innovantes en informatique*, Hermès (collection Informatique et Gestion), 249 p.
- SAINT-JAMES Emmanuel (1993), « Innovations et ruptures dans l'économie et l'épistémologie de l'informatique », communication au *Troisième colloque « Histoire de l'informatique »* organisé par l'Institut National de Recherche en Informatique et Automatique.
- SALAIIS Robert (1989), « L'analyse économique des conventions de travail », in *Revue Economique*, vol. 40, n° 2, pp. 199-240.
- SALAIIS Robert (1994), « Incertitude et interactions de travail : des produits aux conventions », pp. 371-403, in ORLEAN André (éd.), *Analyse économique des conventions*, PUF (Economie), 403 p.
- SALAIIS Robert (éd.) (1998 A), *Institutions et conventions : la réflexivité de l'action économique*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Raisons pratiques 9), 303 p.
- SALAIIS Robert (1998 B), A la recherche du fondement conventionnel des institutions, pp. 255-293, in SALAIIS Robert (éd.).
- SALAIIS Robert et STORPER Michael (1993), *Les mondes de production : enquête sur l'identité économique de la France*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Civilisations et sociétés), 467 p.
- SALAIIS Robert, THEVENOT Laurent (eds.) (1986), *Le travail : marchés, règles, conventions*, Economica / INSEE, 370 p.
- SALZMAN Claude (1982), « Qui dépense quoi ? Les plus rentables ne sont pas toujours ceux que l'on pense », in *01 Informatique*, n° 157, pp. 58-66.
- SAVIOTTI P.P., METCALFE J.S. (1984), « A theoretical approach to the construction of technological output indicators », in *Research Policy*, 13, pp. 141-151.
- SCHILLER Herbert I. (1997), « Les technologies de l'information sous influence : l'instauration d'une ingouvernabilité mondiale », pp. 43-58, in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- SCHREYER Paul (1998), « Information and communication technology and the measurement of real output, final demand and productivity », in *STI Working Papers*, OECD.
- SCHWUCHOW Werner (1997), « L'information : bien public ou privé ? », pp. 113-123 in MAYERE Anne (éd.), *La société informationnelle*, L'Harmattan (Communication et civilisation), 239 p.
- SEN Amartya (1993), *Ethique et économie, et autres essais*, Presses Universitaires de France (Philosophie morale), 364 p.

- SECRETARIAT D'ETAT A L'INDUSTRIE (1999), *Technologies et société de l'information : Etat des lieux statistique*, Service des Etudes et des Statistiques Industrielles, SESSI SJTI / INSEE, 189 p.
- SHANNON Claude E., WEAVER Warren (1949), *The Mathematical Theory of Communication*, University of Illinois Press, 125 p.
- SHAPIRO Carl, VARIAN Hal R. (1999), *Economie de l'information : Guide stratégique de l'économie des réseaux*, De Boeck Université (Ouvertures économiques), 313 p.
- SIMON Herbert Alexander (1974), *La science des systèmes, science de l'artificiel*, EPI, 159 p.
- SIMON Herbert Alexander (1978), « Rationality as a Process and as a Product of Thought », in *American Economic Review*, vol. 68, n° 1, pp. 1-16.
- SIMON Herbert Alexander (1980), *Le nouveau management : la décision par les ordinateurs*, Economica, 159 p.
- SIMON Herbert Alexander (1986), « The Steam Engine and the Computer : What Makes Technology Revolutionary ? », in *Educom Bulletin*, n°22, pp. 2-5.
- SIMON Herbert Alexander (1992), « Programs as factors of production », in *Behavioral Economics and Business Organization*, The MIT Press.
- SMETS-SOLANES Jean-Paul (1998), *L'économie du logiciel libre. Un point de vue d'utilisateur*, <http://www.mmedium.com/dossiers/libre/>.
- SMETS-SOLANES Jean-Paul, FAUCON Benoît (1999), *Logiciels libres : Liberté, égalité, business*, Edispher, 250 p.
- SOBERMAN Marcel (1992), *Génie logiciel en informatique de gestion*, Eyrolles, 243 p.
- SPENCE M. (1984), « Cost Reduction, competition and industry performance », in *Econometrica*, 52.
- STALK G., EVANS P., SHULMAN L. E. (1992), « Competing on Capabilities : The New Rules of Corporate Strategy », in *Harvard Business Review*, March-April, pp. 57-69.
- STALLMAN Richard M. (1998), « Qu'est-ce que le logiciel libre ? », in *FSF & GNU*
- STALLMAN Richard M. (1999), « The GNU Operating System and the Free Software Movement », in DIBONA, Chris, OCKMAN Sam, STONE Mark, *Open sources : voices from the open source revolution*, O'Reilly & Associates, 272 p.
- STEINER Philippe (1999), *La sociologie économique*, La Découverte (Repères), 122 p.
- STEKKE A. (1996) *Economie de l'information et radicalisation de la démocratie : un bref commentaire*, ronéo, 5 p.
- STIGLER George (1951), « The Division of Labor is limited by the Extent of the Market », in *Journal of Political economy*, vol. 59 n°3, Juin, pp. 185-193.
- STIGLITZ Joseph (1985), « Information and economic analysis : a perspective », in *Economic Journal*, 95, pp. 21-41.
- STRASSMANN Paul A. (1990), *The Business Value of Computers*, Information Economics Press, 522 p.
- STRASSMANN Paul A. (1997 A), « Will big spending on computers guarantee profitability ? », in *Datamation*, February, pp. 73-85.
- STRASSMANN Paul A. (1997 B), *The squandered computer : evaluating a business alignment of information technologies*, Information Economics Press, 426 p.
- TARONDEAU Jean-Claude (1998), *Le management des savoirs*, PUF (Que sais-je ?), 127 p.
- TEECE David (1986), « Profiting from technological innovation : implications for integration, collaboration, licensing and public policy », in *Research Policy*, vol. 15, pp. 286-305.
- THEVENOT Laurent (1989), « Equilibre et rationalité dans un univers complexe », in *Revue Economique*, 40 (2), pp. 147-197.
- THEVENOT Laurent (1997), « Un gouvernement par les normes. Pratiques et politiques des formats d'information », pp. 205-242, in CONEIN Bernard, THEVENOT Laurent, *Cognition et information en société*, Editions de l'Ecole des Hautes Etudes en Sciences Sociales (Raisons pratiques 8), 317 p.
- THUROW Lester (1987), « Economic paradigms and slow American productivity growth », in *Eastern Economy Journal*, vol. 13, pp. 333-343.

- TIROLE Jean (1988), *The theory of industrial organization*, MIT Press, 479 p.
- TOFFLER Alvin (1990), *Powershift : knowledge, wealth and violence at the edge of the 21st century*, Bantam Books, 585 p.
- TORRES Asdrad (1996), « A la recherche d'un nouveau souffle industriel », in *Problèmes Economiques* n° 2464-2465, p.14-16, reproduction de l'article d'*Alternatives Economiques* (janvier 1995), paru sous le titre, « Le grand mythe du multimédia ».
- TOURAINÉ Alain (1962), « L'organisation professionnelle de l'entreprise », pp. 387-429, in FRIEDMANN Georges, NAVILLE Pierre, *Traité de sociologie du travail*, Armand Colin, 468 p.
- TREILLE Jean-Michel (1973), *L'économie mondiale de l'ordinateur*, Seuil (Economie et société), 205 p.
- TURKLE Sherry (1986), *Les enfants de l'ordinateur*, Denoël (Présence de la science), 318 p.
- VALLES Ramon Salvador, VAN WUNNIK Lucas, PINEDA Félix (1997), « Potentialités et limites de l'EDI : analyse des entreprises d'un réseau du secteur automobile », pp. 133-151, in MONNOYER-LONGE Marie-Christine (éd.), *L'entreprise et l'outil informationnel*, L'Harmattan, 220 p.
- VELTZ Pierre, ZARIFIAN Philippe (1994), « Travail collectif et modèles d'organisation de la production », in *Le travail humain*, tome 57, n° 3, pp. 239-249.
- VINCENT André L.A. (1968), *La mesure de la productivité*, Dunod (Sigma), 303 p.
- VIRILIO Paul (1998), *La bombe informatique*, Galilée (l'Espace critique), 159 p.
- VIRNO Paolo (1992), « Quelques notes à propos du "general intellect" », in *Futur Antérieur*, n°10.
- VYSSOTSKY V.A. (1973), « Common sense in designing testable software », pp.41-47, in HETZEL W.C., *Program Test Methods*, Prentice Hall.
- WALTON Richard Eaton (1989), *Up and running : integrating information technology and the organization*, Harvard Business Press, 231 p.
- WANG S. (1992), « Contracting for software development », in *Management Science*, vol. 38, n° 3.
- WEINSTEIN Olivier (1989), « Production et circulation des connaissances scientifiques et technologiques », pp. 67-93, in MOULAERT Frank (éd.), *La production des services et sa géographie*, Numéro Spécial de la Revue « Cahiers Lillois d'Economie et de Sociologie », Université de Lille 1, 265 p.
- WEINSTEIN Olivier (1992), « High technology and flexibility », pp. 19-39, in COOKE Philip (et al.) (1992), *Towards global localization : The computing and telecommunications industries in Britain & France*, UCL Press, 227 p.
- WEISSBERG Daniel (1992) *Le système mondial de l'informatique : Acteurs et enjeux*, Thèse de doctorat, Toulouse 2, 415 p.
- WIENER Lauren Ruth (1994), *Les avatars du logiciel : leçons à tirer de quelques jolis fiascos informatiques*, Addison-Wesley (Mutations technologiques), 276 p.
- WILLIAMSON Oliver Eaton (1975), *Markets and hierarchies : analysis and antitrust implications, a study in the economics of internal organization*, The Free Press, 286 p.
- WILLIAMSON Oliver Eaton (1981), « The modern corporation : origins, evolution, attributes », in *Journal of Economic Literature*, vol. XIX, pp. 1537-1568.
- ZIMMERMANN Jean-Benoît (1995 A), « Le concept de grappes technologiques : Un cadre formel », in *Revue Economique*, vol. 46, n° 5, pp. 1263-1295.
- ZIMMERMANN Jean-Benoît (1995 B), « L'industrie du logiciel : de la protection à la normalisation », pp. 195-207, in BASLE Maurice (et al.), *Changement institutionnel et changement technologique : évaluation, droits de propriété intellectuelle, système national d'innovation*, CNRS Editions, 333 p.
- ZIMMERMANN Jean-Benoît (1998 A), «L'industrie du logiciel : ébauche d'une approche prospective», in *Terminal, Technologie de l'Information, Culture et Société* (hiver 97-98), pp. 129-142.
- ZIMMERMANN Jean-Benoît (1998 B), «Un régime de droit d'auteur : la propriété intellectuelle du logiciel», in *Réseaux* n° 88-89, mars-juin, pp. 129-142.

Références bibliographiques

ZIMMERMANN Jean-Benoît (1999), «Logiciel et propriété intellectuelle : du Copyright au Copyleft», *revue Terminal, Technologie de l'Information, Culture et Société*, n° 80/81, « Les logiciels libres : de l'utopie au marché », pp.149-166.

ZUSCOVITCH Ehud (1983), «Informatisation, flexibilité et division du travail », in *Revue d'Economie Industrielle* n° 25, pp. 50-61.

« Le développement des sociétés de services en informatique de 70 à 83 », in *Economie et prévision* (1986) n° 72.