55376
1977
1

N° d'ordre : 202

55376
1977
1

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir le grade de

DOCTEUR INGENIEUR

par

Bernard DEGRENDEL

Ingénieur I.S.E.N.

COMMUNICATIONS ET PROTOCOLES DANS UNE ARCHITECTURE MULTI-MICRO-PROCESSEUR

Thèse soutenue le 27 avril 1977 devant la commission d'examen

MM. P. BACCHUS	Président	
G. BOULAYE	Examinateurs	
C. CARREZ		
L. POUZIN		
V. CORDONNIER	Rapporteur	

030 020648 8

IVERSITE DES SCIENCES
TECHNIQUES DE LILLE

01.02.1976

DOYENS HONORAIRES de l'Ancienne Faculté des Sciences

MM. R. DEFRETIN, H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES des Anciennes Facultés de Droit et Sciences Economiques, des Sciences et des Lettres

M. ARNOULT, Mme BEAUJEU, MM. BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, CORSIN, DEHEUVELS DEHORS, DION, FAUVEL, FLEURY, P. GERMAIN, HEIM DE BALSAC, HOCQUETTE, KAMPE DE FERIET, KOUGANOFF, LAMOTTE, LASSERRE, LELONG, Mme LELONG, MM. LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, NORMANT, PEREZ, ROIG, ROSEAU, ROUBINE, ROUELLE, SAVART, WATERLOT, WIEMAN, ZAMANSKI.

PRESIDENTS HONORAIRES DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. R. DEFRETIN, M. PARREAU.

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

M. J. LOMBARD.

PROFESSEURS TITULAIRES

M.	BACCHUS Pierre	Astronomie
M.	BEAUFILS Jean-Pierre	Chimie Physique
M.	BECART Maurice	Physique Atomique et Moléculaire
M.	BILLARD Jean	Physique du Solide
M.	BIAYS Pierre	Géographie
M.	BONNEMAN Pierre	Chimie Appliquée
M.	BONNOT Ernest	Biologie Végétale
M.	BONTE Antoine	Géologie Appliquée
M.	BOUGHON Pierre	Algèbre
M.	BOURIQUET Robert	Biologie Végétale
M.	CELET Paul	Géologie Générale
M.	CONSTANT Eugène	Electronique
M.	DECUYPER Marcel	Géométrie
M.	DELATTRE Charles	Géologie Générale
M.	DELHAYE Michel	Chimie Physique
M.	DERCOURT Michel	Géologie Générale
M.	DURCHON Maurice	Biologie Expérimentale
M.	FAURE Robert	Mécanique
M.	FOURET René	Physique du Solide
M.	GABILLARD Robert	Electronique
M.	GLACET Charles	Chimie Organique
M.	GONTIER Gérard	Mecanique
M.	GRUSON Laurent	Algèbre
M.	GUILLAUME Jean	Microbiologie
M.	HEUBEL Joseph	Chimie Minérale
M.	LABLACHE-COMBIER Alain	Chimie Organique
M.	LANSRAUX Guy	Physique Atomique et Moléculaire
M.	LAVEINE Jean-Pierre	Paléontologie
M.	LEBRUN André	Electronique
M.	LEHMANN Daniel	Géométrie

Mme	LENOBLE Jacqueline	Physique Atomique et Moléculaire
M.	LINDER Robert	Biologie et Physiologie Végétales
M.	LOMBARD Jacques	Sociologie
M.	LOUCHEUX Claude	Chimie Physique
M.	LUCQUIN Michel	Chimie Physique
M.	MAILLET Pierre	Sciences Economiques
M.	MONTARIOL Frédéric	Chimie Appliquée
M.	MONTREUIL Jean	Biochimie
M.	PARREAU Michel	Analyse
M.	POUZET Pierre	Analyse Numérique
M.	PROUVOST Jean	Minéralogie
M.	SALMER Georges	Electronique
M.	SCHILTZ René	Physique Atomique et Moléculaire
Mme	SCHWARTZ Marie-Hélène	Géométrie
M.	SEGUIER Guy	Electrotechnique
M.	TILLIEU Jacques	Physique Théorique
M.	TRIDOT Gabriel	Chimie Appliquée
M.	VIDAL Pierre	Automatique
M.	VIVIER Emile	Biologie Cellulaire
M.	WERTHEIMER Raymond	Physique Atomique et Moléculaire
M.	ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS SANS CHAIRE

M.	BELLET Jean	Physique Atomique et Moléculaire
M.	BODARD Marcel	Biologie Végétale
M.	BOILLET Pierre	Physique Atomique et Moléculaire
M.	BOILLY Bénoni	Biologie Animale
M.	BRIDOUX Michel	Chimie Physique
M.	CAPURON Alfred	Biologie Animale
M.	CORTOIS Jean	Physique Nucléaire et Corpusculaire
M.	DEBOURSE Jean-Pierre	Gestion des entreprises
M.	DEPREZ Gilbert	Physique Théorique
M.	DEVRAINNE Pierre	Chimie Minérale
M.	GOUDMAND Pierre	Chimie Physique
M.	GUILBAULT Pierre	Physiologie Animale
M.	LACOSTE Louis	Biologie Végétale
Mme	LEHMANN Josiane	Analyse
M.	LENTACKER Firmin	Géographie
M.	LOUAGE Francis	Electronique
Mle	MARQUET Simone	Probabilités
M.	MIGEON Michel	Chimie Physique
M.	MONTEL Marc	Physique du Solide
M.	PANET Marius	Electrotechnique
M.	RACZY Ladislas	Electronique
M.	ROUSSEAU Jean-Paul	Physiologie Animale
M.	SLIWA Henri	Chimie Organique

MAITRES DE CONFERENCES (et chargés d'Enseignement)

M.	ADAM Michel	Sciences Economiques
M.	ANTOINE Philippe	Analyse
M.	BART André	Biologie Animale
M.	BEGUIN Paul	Mécanique
M.	BKOUCHE Rudolphe	Algèbre
M.	BONNELLE Jean-Pierre	Chimie
M.	BONNEMAIN Jean-Louis	Biologie Végétale
M.	BOSCQ Denis	Probabilités
M.	BREZINSKI Claude	Analyse Numérique
M.	BRUYELLE Pierre	Géographie

.../...

M. CARREZ Christian	Informatique
M. CORDONNIER Vincent	Informatique
M. COQUERY Jean-Marie	Psycho-Physiologie
Mle DACHARRY Monique	Géographie
M. DEBENEST Jean	Sciences Economiques
M. DEBRABANT Pierre	Géologie Appliquée
M. DE PARIS Jean-Claude	Mathématiques
M. DHAINAUT André	Biologie Animale
M. DELAUNAY Jean-Claude	Sciences Economiques
M. DERIEUX Jean-Claude	Microbiologie
M. DOUKHAN Jean-Claude	Physique du Solide
M. DUBOIS Henri	Physique
M. DYMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du Solide
Me EVRARD Micheline	Chimie Appliquée
M. FONTAINE Jacques-Marie	Electronique
M. FOURNET Bernard	Biochimie
M. FORELICH Daniel	Chimie Physique
M. GAMBLIN André	Géographie
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GRANELLE Jean-Jacques	Sciences Economiques
M. GUILLAUME Henri	Sciences Economiques
M. HECTOR Joseph	Géométrie
M. JACOB Gérard	Informatique
M. JOURNEL Gérard	Physique Atomique et Moléculaire
Mle KOSMAN Yvette	Géométrie
M. KREMBEL Jean	Biochimie
M. LAURENT François	Automatique
Mle LEGRAND Denise	Algèbre
Mle LEGRAND Solange	Algèbre
M. LEROY Jean-Marie	Chimie Appliquée
M. LEROY Yves	Electronique
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique Théorique
M. LOUCHET Pierre	Sciences de l'Education
M. MACKE Bruno	Physique
M. MAHIEU Jean-Marie	Physique Atomique et Moléculaire
Me N'GUYEN VAN CHI Régine	Géographie
M. MAIZIERES Christian	Automatique
M. MALAUSSENA Jean-Louis	Sciences Economiques
M. MESSELYN Jean	Physique Atomique et Moléculaire
M. MONTUELLE Bernard	Biologique Appliquée
M. NICOLE Jacques	Chimie Appliquée
M. PAQUET Jacques	Géologie Générale
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie Physique
M. PERROT Pierre	Chimie Appliquée
M. PERTUZON Emile	Physiologie Animale
M. PONSOLLE Louis	Chimie Physique
M. POVY Lucien	Automatique
M. RICHARD Alain	Biologie
M. ROGALSKI Marc	Analyse
M. ROY Jean-Claude	Psycho-Physiologie
M. SIMON Michel	Sociologie
M. SOMME Jean	Géographie
Mle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. STEEN Jean-Pierre	Informatique

.../...

M.	THERY Pierre	Electronique
M.	TOULOTTE Jean-Marc	Automatique
M.	TREANTON Jean-René	Sociologie
M.	VANDORPE Bernard	Chimie Minérale
M.	VILLETTE Michel	Mécanique
M.	WALLART Francis	Chimie
M.	WERNIER Georges	Informatique
M.	WATERLOT Michel	Géologie Générale
Mme	ZINN-JUSTIN Nicole	Algèbre

Je tiens à remercier Monsieur le Professeur P. Bacchus qui me fait l'honneur de présider le jury de cette thèse.

Je suis très reconnaissant envers Monsieur le Professeur V. Cordonnier qui m'a proposé ce travail et m'a guidé de ses conseils.

Je remercie Messieurs les Professeurs G. Boulaye et C. Carrez ainsi que Monsieur L. Pouzin, Directeur des projets pilotes de l'IRIA, qui se sont intéressés à ce travail et ont accepté de participer au jury de cette thèse.

Monsieur le Professeur A. Lebrun, Directeur du C.U.E.E.P., a montré une grande compréhension et a facilité la poursuite de ce travail ; qu'il soit assuré de ma gratitude.

Que Messieurs A. Arnold, G. Cany, B. Toursel soient remerciés pour les discussions constructives et amicales que nous avons menées sur ce travail. En particulier, A. Arnold a apporté des idées précieuses sur les protocoles de communication.

Je tiens également à remercier Mademoiselle Françoise Desquiens qui a assuré avec compétence et gentillesse la dactylographie de cette thèse, ainsi que Madame et Monsieur P. Debock qui ont assuré dans les meilleures conditions la réalisation de ce document.

TABLE DES MATIERES

INTRODUCTION

CHAPITRE I : TRAITEMENTS ET STRUCTURES DE MACHINES

- IA : LES SYSTEMES DE CONTROLE ET DE COMMANDE
- IB : ARCHITECTURES PARALLELES
- IC : UN PREMIER CHOIX D'ARCHITECTURE
- ID : LA GESTION DE TRANSACTION
- IE : CONCLUSION

CHAPITRE II : LE RÉSEAU DE COMMUNICATIONS

- IIA : INTRODUCTION
- IIB : DESCRIPTION DU PROCESSEUR CELLULAIRE DE COMMUNICATIONS
- IIC : ETUDE DE QUELQUES ASPECTS COMPLEMENTAIRES DU P.C.C.
- IID : MODELISATION DU P.C.C.
- IIE : NOUVELLE DEFINITION DU PROCESSEUR DE COMMUNICATIONS
- IIF : DEGRADATION DE L'ETAT DE P.R.O.C.O.P.E.
- IIG : CONCLUSION

CHAPITRE III : PROTOCOLES DE COMMUNICATIONS

- IIIA : GENERALITES
- IIIB : HIERARCHIE DE MACHINES
- IIIC : PRINCIPES GENERAUX REGISSANT LES COMMUNICATIONS DANS UN SYSTEME MULTIPROCESSEUR
- IIID : MISE EN OEUVRE DES PROTOCOLES

CHAPITRE IV : ARCHITECTURE ET SYSTÈME D'EXPLOITATION

- IVA : INTRODUCTION
- IVB : CHOIX D'UNE ARCHITECTURE
- IVC : STRUCTURE GENERALE DU SYSTEME D'EXPLOITATION
- IVD : FONCTION DE SURVEILLANCE
- IVE : FONCTION ALLOCATRICE

CONCLUSION

INTRODUCTION

Il paraît évident d'affirmer que les performances d'un système de traitement de l'information dépendent essentiellement de trois facteurs : l'application ou le type d'application considéré, l'outil informatique (dans ses aspects matériels et logiciels). L'adaptation entre l'application et l'outil informatique, Cette simple constatation générale est d'autant plus vraie que l'outil informatique a une structure complexe. En particulier les multiprocesseurs posent, en plus des problèmes connus dans les monoprocesseurs, des problèmes nouveaux liés au nombre d'unités et donc en grande partie des problèmes de communications.

Ces considérations sont à la base de ce travail et ont guidé la démarche suivie. Afin d'adapter l'application et l'outil informatique, la meilleure approche nous a semblé être de définir l'outil grâce aux caractéristiques de l'application. Parmi les nombreuses raisons citées en faveur des systèmes multiprocesseurs (travaux excédant les possibilités d'un monoprocesseur, structure de certains types de travaux, fiabilité accrue, partage de ressources, multiplicité des utilisateurs, etc ...) nous avons tenté d'en étudier certaines plus précisément en nous attachant à un type particulier d'application : la gestion transactionnelle. Cette étude (Chapitre I) permet de dégager des grandes lignes caractérisant l'outil adapté. Etant donnée l'importance des

communications, tant sur le plan de l'outil physique à utiliser que sur celui des règles d'utilisation de cet outil et des interactions entre les unités reliées, les chapitres II et III sont respectivement consacrés au réseau de communications et aux protocoles de communications. Partant des spécifications posées au chapitre I et des conclusions des chapitres II et III, le chapitre IV est consacré à une définition de l'outil informatique du point de vue des unités connectées et de l'utilisation de ces unités (bases d'un système d'exploitation).

I

TRAITEMENTS ET STRUCTURES DE MACHINES

IA LES SYSTÈMES DE CONTRÔLE ET DE COMMANDE.

Un système de contrôle et de commande [AN62] est un système informatique capable de gérer et diriger tous les aspects du fonctionnement d'un vaste complexe d'hommes et (ou) de machines. Par exemple, certaines applications militaires, le contrôle de l'espace aérien, certains processus industriels complexes sont des systèmes de contrôle et de commande (S.C.C.).

Le fonctionnement d'un S.C.C. se caractérise par une grande quantité de travaux simultanés, divers mais interdépendants, à effectuer immédiatement. Ils sont exécutés au mieux par un système centralisé et intégré de traitement de données. Les travaux externes au système étant parallèles par nature, il est souhaitable que les traitements internes au S.C.C. puissent se faire en parallèle.

Les principales caractéristiques requises pour un tel système sont : la disponibilité, l'adaptativité, l'expansibilité, la facilité de programmation.

IA1 LA DISPONIBILITE D'UN S.C.C. .

Pour un utilisateur, c'est la proportion du temps de fonctionnement disponible sur une période donnée. En général, à la conception d'un système informatique il faut comparer le coût de l'indisponibilité à celui des méthodes à mettre en oeuvre pour la combattre (fiabilité du matériel,

banalisation, redondances, ...) ; les S.C.C. présentent un coût d'indisponibilité très élevé justifiant les investissements nécessaires à la prévenir. Selon le type d'application et ses exigences, il faudra qu'une plus ou moins grande proportion du système soit toujours utilisable pour les fonctions vitales et que tout le système soit disponible la plupart du temps. Ceci exclut pratiquement les arrêts prolongés pour maintenance et nécessite un matériel très fiable et de maintenance aisée. Afin de détecter les pannes et d'en réduire les conséquences, le système doit à tout moment être en mesure de tester et évaluer le fonctionnement de certaines voire toutes ses parties (logicielles et matérielles). Durant ces mesures, les possibilités du S.C.C. doivent être peu altérées et il doit pouvoir retourner rapidement à un fonctionnement aussi normal que possible. La disponibilité est à réaliser soit en fonctionnement normal, ce qui revient essentiellement à assurer le service en temps réel (charge fluctuante, temps de réponse faible), soit en fonctionnement anormal, d'où une notion d'adaptativité.

IA2 L'ADAPTATIVITE D'UN S.C.C. .

Un S.C.C. est adaptatif s'il est capable à tout instant d'analyser les demandes de travail et d'en déterminer les meilleurs choix sur les processus à activer en fonction de la disponibilité de ses composants logiciels et matériels. Quatre catégories d'adaptativité peuvent être distinguées :

- adaptation selon les types de travaux et leurs besoins respectifs (calcul, entrées-sorties, fréquence d'accès mémoire, etc...)

- adaptation selon la charge de travail fournie par les utilisateurs.

- adaptation selon la charge de travail interne ; il s'agit de faire travailler le système au mieux malgré les mesures et tests, malgré la gestion des erreurs logicielles, etc...

- adaptation selon la disponibilité des ressources matérielles. Les ressources matérielles peuvent varier selon les pannes affectant soit les composants (mémoires, processeurs...) soit les liens entre ceux-ci. Il faut donc modifier rapidement, dynamiquement, automatiquement les relations entre processus actifs et ressources allouées.

Cette reconfiguration doit évidemment fournir un système viable fonctionnant en mode dit "dégradé", donc présentant des performances plus faibles qu'en fonctionnement normal. L'écart est d'autant moins sensible que la charge de travail est faible ; la surcharge interne due à la reconfiguration doit donc être la plus brève possible.
Afin de faciliter l'adaptivité il est intéressant de

- fragmenter largement le système pour limiter l'impact d'une panne et faciliter la reconfiguration
- dimensionner largement le système en mettant en oeuvre des unités en grand nombre. Généralement une partie de ces unités est inutilisée mais peut être sollicitée en cas de besoin (charge élevée, reconfiguration).

IA3 L'EXPANSIBILITE DU S.C.C.

Il s'agit de concevoir un système pouvant évoluer en puissance selon les besoins des utilisateurs. Cette caractéristique est souhaitable dans tout système informatique mais l'importance d'un S.C.C. (et donc son coût), sa complexité, le coût de l'indisponibilité la rendent quasi-indispensable dans les S.C.C. .

L'expansibilité doit permettre d'étendre les possibilités du système :

- soit par une augmentation quantitative du matériel existant (cas d'une extension mémoire).

- soit par ajout de ressources auparavant absentes du système (périphériques spéciaux, processeurs spécialisés).

- soit par remplacement de certaines ressources par d'autres, fonctionnellement identiques, qui sont plus performantes, plus disponibles, plus sûres, etc... Les caractéristiques d'accès (vitesse, commandes, ...) à ces nouvelles ressources peuvent être différentes de celles des matériels remplacés. Les ressources substituées peuvent être de toutes les fonctions (entrées-sorties, mémoires, processeurs ...) alors que sur les machines actuelles ne sont remplacés que des équipements périphériques, voire la mémoire centrale mais jamais les processeurs.

Afin de respecter le critère de disponibilité, les refontes complètes et fréquentes du système d'exploitation sont à proscrire ; l'expansibilité doit donc être aussi adaptative que possible.

IA4 PROGRAMMATION DES S.C.C. .

Elle doit être compatible avec les caractéristiques précédentes, ce qui signifie :

- que l'expansion du S.C.C. doit entraîner le minimum de programmation et donc que celle-ci doit être peu dépendante de la configuration matérielle existante.

- que le programme doit être indépendant de la configuration du S.C.C. lors de l'exécution puisque la reconfiguration est dynamique.

- que le parallélisme doit être défini de façon souple en se donnant des possibilités d'exécution parallèle. Celles-ci seront ou non utilisées selon les ressources disponibles dans le système lors de l'exécution

- qu'éventuellement, suite à une évolution du matériel ou indépendamment de celle-ci, les langages de programmation disponibles varieront.

D'autre part, le volume et la complexité des travaux à exécuter rendent indispensables la modularité de la programmation, ce qui peut poser des contraintes sur le logiciel de base (éditeurs de liens).

Ces caractéristiques posent des problèmes d'organisation et d'architecture, ils ne sont donc pas résolus par la technologie seule (d'autant plus que parallèlement à son progrès croissent les exigences des utilisateurs). La structure monoprocesseur est certes la solution la plus simple, elle ne respecte toutefois pas les contraintes posées.

C'est également le cas de certaines architecture parallèles qui résolvent le problème de la puissance de traitement grâce :

- à une subdivision de l'unité centrale en sous-ensembles spécialisés commandés par une unité commune de contrôle et de commande (C.D.C. Star) [HIN 72].

- à une multiplication des unités arithmétiques et logiques gérées par une unité commune de contrôle et de commande (ILLIAC IV) [MAC 70].

- etc...

De telles structures

1) ne sont pas universelles ; en tous cas elles sont inadaptées à la majorité des applications de gestion.

2) sont souvent trop rigides, i.e. peu aptes à se conformer aux exigences qualitatives et quantitatives des utilisateurs.

3) n'offrent pas des performances nettement supérieures aux machines classiques étant donnés leur coût et leur spécialisation.

4) ont un coût prohibitif (leur fabrication peut se comparer à de la confection "sur mesure" dans une société de "prêt à porter").

IB ARCHITECTURES PARALLÈLES.

Le parallélisme interne au système peut être mis en oeuvre soit par concurrence, soit par simultanéité. Il y a concurrence lorsque divers travaux se déroulent dans un certain intervalle de temps ; il y a simultanéité lorsque divers travaux se déroulent au même instant. Dans le premier cas, le parallélisme n'est qu'apparent et est mis en oeuvre par des techniques de multiprogrammation par exemple ; il pose certes des problèmes au niveau de l'architecture des machines, mais bien moins que le parallélisme "réel" du second cas auquel nous nous sommes intéressés.

Dans les deux cas, le parallélisme peut être défini à différents niveaux :

1) entre ensembles de programmes de différents utilisateurs
2) entre programmes soumis par un même utilisateur
3) entre séquences d'instructions d'un même programme
4) entre instructions d'une même séquence
5) entre phases de préparation et d'exécution d'une instruction
6) entre opérations binaires au cours de l'exécution d'une instruction.

Diverses combinaisons entre ces solutions sont à envisager sur une même machine. Le dernier de ces niveaux est tout à fait commun ; le cinquième est très répandu grâce aux techniques de l'électronique et de la microprogrammation horizontale. Le parallélisme pose donc des problèmes d'architecture aux quatre premiers niveaux. Le parallélisme entre instructions d'une même séquence paraît adapté lorsque le problème à résoudre est très bien défini, nécessite un traitement immédiat et exige peu de synchronisations entre les activités simultanées ; il présente toutefois des difficultés car les circuits de décodage sont complexes et coûteux ; ils risquent d'autre part d'être un facteur de ralentissement. Le parallélisme entre séquences d'instructions (dont un cas particulier est donné par l'exécution parallèle d'opérations d'entrées-sorties et d'opérations de calcul) paraît utilisable pour des traitements répétitifs, dont la structure a été définie dans ce sens par l'analyse, la programmation et la compilation ; une exécution rapide rend cette solution adaptée au "temps réel".

Le parallélisme entre programmes d'un même utilisateur ne présente que peu d'intérêt, soit parce que ces programmes sont indépendants (ce qui revient au premier cas), soit parce qu'ils sont très dépendants et nécessitent une exécution séquentielle. Enfin, le parallélisme entre utilisateurs permet essentiellement de tirer profit d'une grande capacité de traitement disponible.

A ces niveaux de parallélisme correspondent plusieurs niveaux de détection du parallélisme : analyse, programmation, compilation, exécution [FLY70] [KOC69] [BER66].

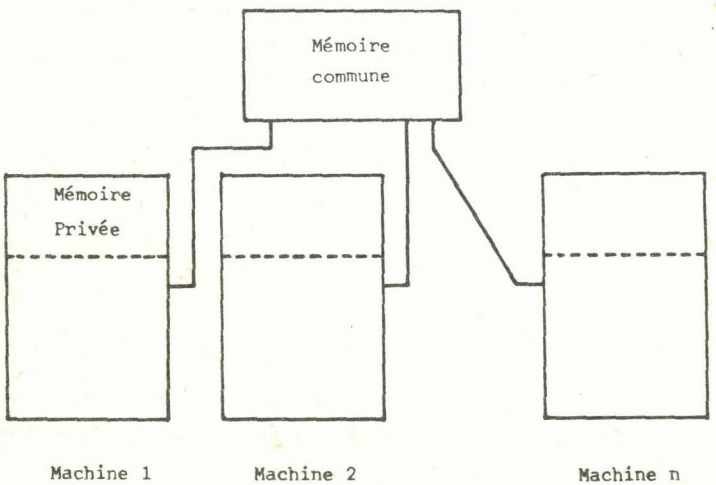
Les architectures sur lesquelles est mis en oeuvre le parallélisme sont : la multiplication de machines classiques, le partage d'une mémoire commune par plusieurs machines classiques, les multiprocesseurs.

IB1 MULTIPLICATION DE MACHINES CLASSIQUES.

Plusieurs exemplaires d'une même machine travaillent simultanément et appliquent donc le parallélisme au premier niveau défini. La puissance de l'ensemble est élevée mais si la disponibilité est possible en théorie vue la multiplicité des moyens, les problèmes de coordination, de partage de données communes, de reconfiguration ... posent des problèmes pratiquement insolubles.

IB2 PARTAGE D'UNE MEMOIRE COMMUNE.

Cette architecture se schématise généralement :



Elle permet le partage de données stockées en mémoire commune tout en réalisant une protection simple sur les mémoires privées des différentes machines. Les performances restent toutefois médiocres, en particulier à cause de la complexité et de la longueur des transferts entre mémoires, de la difficulté d'équilibrer le système (affecter au mieux des ressources à la machine qui en a le plus besoin). La reconfiguration et la coordination des activités ne sont résolues que si la coopération entre processeurs est quasi-nulle, i.e. aux deux premiers niveaux. Elles nécessitent qu'une machine joue le rôle de maître d'où des difficultés en cas de reconfiguration et le risque de sous-utilisation de cette machine "maître".

IB3 MULTIPROCESSEURS.

De nombreuses définitions d'un multiprocesseur foisonnent dans la littérature informatique ; parmi cette variété, voici une définition possible.

IB3a *Définition* [ENS 75]

Un multiprocesseur est un système comprenant au moins deux processeurs de performances et de possibilités comparables. Tous les processeurs partagent l'accès à une mémoire commune (sans préjuger de la structure de celle-ci) et aux moyens d'entrées-sorties. Le système est gérè par un système d'exploitation régissant l'interaction entre les processeurs, les ressources communes, les travaux à exécuter.

Selon la définition du mot "travail", le parallélisme se situera à l'un des trois premiers niveaux défini précédemment.

IB3b Principaux problèmes posés par une architecture de multiprocesseur [HOB] [LEV73] [FLY]

Moyen de communication.

Les différents processeurs communiquent soit entre eux, soit avec des ressources communes. Les communications entre processeurs peuvent se

faire soit directement par échange de messages, soit par l'intermédiaire de mémoires (boîtes aux lettres) mais elles doivent toujours emprunter un moyen de communication permettant soit des transferts séquentiels (monobus, commutateur de voies,...) soit des transferts simultanés (réseau à décalage, multibus, processeur cellulaire de communications, ...) entre les unités qui lui sont connectées.

Le moyen de communication peut véhiculer des quantités élémentaires d'information de formats variables (messages de format fixe ou variable paquets, etc...). Il peut être hiérarchisé en niveaux, hétérogène ou homogène aux divers niveaux de la hiérarchie.

Quelles que soient ses caractéristiques, il est souhaitable

- qu'il ne cause aucun ralentissement important dans l'activité des objets connectés, soit du fait de son débit, soit du fait de sa vitesse de transfert
- qu'il soit transparent totalement aux utilisateurs du système, si possible aux constituants du système.

Le contrôle des communications.

Le contrôle des communications, c'est-à-dire l'ensemble des moyens matériels ou logiciels gérant l'activité du moyen de communication peut présenter des caractéristiques très diverses :

- centralisation des décisions d'acheminement (commutateur de voies) ou décentralisation de ces décisions (réseaux)
- protocole d'échange plus ou moins rigide, plus ou moins complexe.

Contrôle d'exécution d'une séquence d'instructions.

Un processeur peut exécuter une séquence d'instructions soit grâce à sa propre logique de suivi et de contrôle, soit sous les ordres d'une unité de commande commune à plusieurs exécutants (exemple de l'ILLIAC IV).

Cette notion dépend de la définition précise du mot processeur.

Ce contrôle pourra d'ailleurs être différent selon le niveau de langage (microprogramme, langage machine).

Contrôle du système et allocation des tâches.

Pour des raisons d'efficacité, de rapidité, de sécurité, le contrôle du système doit plutôt être décentralisé ; par contre la centralisation permet plus facilement d'éviter les choix contradictoires et les risques d'étreinte fatale. Le choix du système d'exploitation se fait dans cette alternative, chaque terme présentant diverses possibilités de mise en oeuvre (contrôle figé et inamovible, contrôle électif, circulation d'un droit, etc...)

La structure des processeurs et leur nombre.

Ils peuvent être spécialisés (par construction ou par microprogrammation) d'où une grande variété et une grande efficacité, ou relativement universels (interchangeables entre eux) d'où une disponibilité et une adaptativité plus aisées.

Leur structure peut être rudimentaire (unité arithmétique et logique associée à des registres comme dans ILLIAC IV) [MAC70] ou évoluée, voire proche des mini-ordinateurs (système d'interruptions, mémoire locale importante).

La structure de la mémoire.

Celle-ci peut être centralisée en un seul module accessible par les autres constituants du système ou décentralisée en plusieurs modules pourvus éventuellement de règles d'allocation et d'accès.

Une proportion variable peut également être associée aux processeurs (mémoire locale, cache).

L'espace mémoire peut être géré de diverses façons : champ d'adressage unique partagé entre tous les processeurs, champs d'adressage propres à chaque processeur, utilisation de registres de re-allocation etc...

La structure des organes d'entrée-sortie.

Tout comme la mémoire, ceux-ci peuvent former un parc disponible par tous les processeurs, peuvent être associés à un processeur particulier, ou peuvent être répartis parmi les processeurs.

Conclusion.

La multiplicité des choix possibles au sein d'une architecture multiprocesseurs conduit à deux conséquences :

- il est utile de mettre en évidence des critères de classification aussi généraux et universels que possible.
- le choix d'une configuration parmi de multiples options relève de l'art de l'architecte. Il est cependant nécessaire d'appuyer ces choix par des considérations relatives aux performances dans un domaine d'application grâce à des outils d'évaluation aussi précis et fidèles que possible.

IB3c Classification des systèmes multiprocesseurs.

Les systèmes multiprocesseurs sont généralement conçus en vue d'une application particulière, leur classification est donc délicate étant donné leurs propriétés spécifiques.
Plusieurs classifications ont été proposées :

Classification par flots.

Les flots de données et d'instructions circulant dans la machine ont amené Flynn à proposer la classification suivante [FLY72].

- S·I·S·D : 1 flot de données et 1 flot d'instructions. C'est la structure d'une machine classique à tout instant.

- S·I·M·D : 1 flot d'instructions agissant sur plusieurs flots de données. C'est la structure de machines telles que SOLOMON et ILLIAC IV [MAC70].

- M·I·S·D : 1 flot de données soumis à plusieurs flots d'instructions. Peu de machines ont véritablement cette structure.

- M·I·M·D : Plusieurs flots de données interagissent avec plusieurs flots d'instructions. Cette structure, la plus complexe, a connu peu de réalisations originales ; c'est celle de la machine de Holland [HOL59], du réseau de Koczela [KOC69], de la machine de Flynn [FLY].

Autres classifications [HOB] [SEA75].

a) D'après les ressources mises en parallèle (flots de données, fonctions ou unités de contrôle, unités de traitement fonctionnelles).

b) D'après les réalisations

- multi-ordinateurs et multi-processeurs (Multics, Pilot ...)
- processeurs associatifs (DLM, Good Year)
- réseaux et tableaux de processeurs (ILLIAC IV)
- machines fonctionnelles (Gamma 60).

c) D'après la structure des processeurs

- machines à processeurs fonctionnels spécialisés par construction ou par microprogrammation figée

- machines à processeurs fonctionnels non spécialisés ou à spécialisation non figée (mémoire de microprogrammes rechargeable).

Ces classifications restent toutefois arbitraires et peu précises, tandis que la classification de Flynn est simple, quasi indépendante de la réalisation mais est trop générale.

IC UN PREMIER CHOIX D'ARCHITECTURE.

Les caractéristiques des S.C.C. précédemment décrites conduisent à certaines conclusions quant à une architecture adaptée ; certaines de ces conclusions sont relativement générales, d'autres sont précises.

Une architecture supportant un S.C.C. est de préférence caractérisée par :

- une architecture multiprocesseurs
- un nombre élevé (10 ou plus) de processeurs (unité de contrôle et de commande + organes d'exécution) afin de garantir une dégradation progressive
- un processeur possèdera une mémoire locale (nécessaire en cas de tests ou de procédures de recouvrement d'erreur autonomes) de taille limitée (afin de faciliter les reconfigurations) lui conférant une certaine autonomie
- une mémoire accessible par tous les processeurs, de préférence répartie en plusieurs modules (fiabilité, simultanéité d'accès)
- des organes et des processeurs d'entrées-sorties accessibles à tous les processeurs
- la mémoire sera gérée
  - sans interfoliation entre les modules
  - en plusieurs champs d'adressage propres aux différents processeurs
- le couplage (qui traduit la nature, la fréquence, l'importance des interactions entre processeurs) sera d'une nature aussi élevée que possible (par exemple des mécanismes décrits par Patil dans [PAT75] sont écartés) et de fréquence limitée

- il se fera grâce à un moyen de communications distinct des outils de traitement (d'où une simplification de l'architecture)
  - au contrôle décentralisé
  - dont le graphe permettra toutes les connexions désirables
  - transparent aux caractéristiques des objets
  - extensible afin de permettre l'évolution du système

- étant donné le coût des processeurs, [MAI73] il sera préférable d'utiliser des micro-processeurs, ce qui permettra pour un même coût global (conception + réalisation) qu'avec des unités spécifiques d'obtenir un meilleur service. En particulier, leur prix réduit permet de "gaspiller" certaines ressources (processeurs) au profit d'une simplification de l'architecture (d'où un gain en fiabilité, en maintenabilité, voire en vitesse) et du système d'exploitation (si le coût du système d'exploitation est de l'ordre de celui des ressources qu'il gère, mieux vaut augmenter leur nombre et le simplifier) [SEA 75] [FUL73].

L'utilisation de micro-processeurs permet également la réalisation d'unités spécialisées (programmation figée ou mémoire de micro-programme rechargeable) et la micro-programmation de macro-instructions d'où un gain en performance, en fiabilité, en coût de programmation (tendance vers des machines à langage de haut niveau).

Afin de préciser davantage cette architecture (en particulier le niveau de parallélisme), la section suivante aborde l'étude de la gestion de transactions, domaine d'applications dont les S.C.C. font le plus souvent partie ou sont proches.

ID LA GESTION DE TRANSACTIONS.

ID1 INTRODUCTION.

La proportion des applications informatiques de gestion des entreprises (comptabilité, gestion des stocks, statistiques de ventes, etc...) est souvent estimée à 80 % au moins du volume total des traitements

informatiques. Une catégorie particulière d'applications, la gestion de transactions, a acquis une grande importance causée par :

- les progrès des télécommunications tant en quantité, qu'en qualité, qu'en coût
- les progrès des ordinateurs en vitesse de traitement, en capacité, en rapport performance/prix
- les progrès réalisés dans la conception, la réalisation, la commercialisation des unités périphériques
- le désir de plus en plus marqué des gestionnaires, comptables et autres responsables de connaître quasi instantanément la situation de leur entreprise. Ce besoin d'apprécier l'état d'une entreprise n'est pas nouveau ; deux raisons en rendent toutefois l'informatisation utile : la complexité et la taille des entreprises. Leurs origines sont multiples (concentration de sociétés, croissance économique, décentralisation des activités des entreprises, etc...) et elles entraînent inévitablement la manipulation de grands volumes d'information. De plus, ce besoin doit être satisfait rapidement afin d'adapter la gestion et les décisions à la situation de l'entreprise la plus précise possible (i.e. non altérée par le temps). La situation à appréhender peut se situer tant au niveau microscopique (état d'un compte, d'un stock, ...) qu'au niveau macroscopique (balances, statistiques, ...).

Ces facteurs sont certes interdépendants, les progrès entraînant de plus grandes exigences d'où de nouvelles réalisations. Leur convergence vers la gestion de transactions en temps réel conduit à des systèmes dont les caractéristiques sont voisines de celles des S.C.C. . L'objet de cette partie est donc de préciser la notion de gestion de transactions et d'en cerner les problèmes afin de mieux définir ce que doit comporter une architecture adaptée.

ID2 QUELQUES EXEMPLES.

Les applications les plus importantes et les plus évidentes de la gestion de transactions (G.T.) se trouvent en gestion des entreprises.

La liste suivante donne un aperçu de la diversité des secteurs et applications concernés :

- tenue de comptes-clients et applications bancaires
- réservation de places (trains, avions, agences de voyage)
- tenue des stocks
- nomenclatures industrielles et systèmes de tenue de dossiers (médecine)
- assistance au marketing
- gestion d'un ensemble de produits tels que des logements libres d'occupation
- facturation
- etc ...

Comme indiqué ci-dessus, les S.C.C. sont souvent proches des systèmes de G.T., par exemple la gestion de l'espace aérien, le contrôle de processus industriels.

ID3 DEFINITION ET PROBLEMES POSES PAR LA G.T.

ID3a Définition et généralités[GRO73]

Caractéristiques générales.

Un système de gestion de transactions est un système informatique de gestion organisé autour d'une base de données.

Ce système doit permettre simultanément à plusieurs utilisateurs d'en exploiter et d'en manipuler les informations en temps réel (à l'échelle des utilisateurs).

Les utilisateurs sont reliés à une machine informatique grâce à des terminaux ; ce ne sont pas des spécialistes de l'informatique (à la différence de certains systèmes de temps partagé dont les utilisateurs sont des scientifiques plus ou moins compétents).

Définition d'un S.G.T. [GR073]

Un système de gestion de transactions (S.G.T.) est un système informatique où :

- *l'application d'une excitation (ou événement)*
  c'est-à-dire le plus souvent d'une commande accompagnée d'une donnée ou information à traiter ; de manière plus générale c'est l'occurrence d'une ressource,
- *provoque l'exécution d'un travail fini, réduit, autonome*
  (i.e. n'interagissant pas directement avec d'autres transactions),
- *faisant référence à des ensembles de données stockées sur des mémoires de masse*
- *dont le résultat doit être disponible en temps réel ;*
  c'est-à-dire quasi immédiatement à l'échelle humaine (quelques secondes à quelques dizaines de secondes).

Pour compléter cette définition, il faut souligner que le nombre d'utilisateurs connectés au système peut être très élevé, allant jusqu'à quelques milliers ; le nombre de transactions en cours à un instant donné peut être de plusieurs dizaines.

Quantification de quelques éléments d'une transaction.

L'excitation : l'ensemble des informations la constituant est de l'ordre de quelques dizaines à une centaine de caractères.

Le travail : dans les cas les plus simples, l'unité centrale exécute quelques milliers d'instructions ; dans des cas plus complexes ce sera quelques dizaines de milliers d'instructions. Il est à noter que ces instructions comprennent peu de calculs mathématiques complexes (fonctions, virgule flottante).

Les références aux données : leur nombre varie de quelques unités à quelques dizaines. Le volume de données est considérable.

Le résultat : son volume est comparable à celui de l'excitation et varie souvent de quelques dizaines à quelques centaines de caractères.

Ces ordres de grandeur permettent de noter que le système d'exploitation d'un S.G.T. doit être simple et rapide a exécuter car les travaux ont un volume individuel faible et y font souvent appel.

Principaux problèmes posés par la G.T.

D'après les paragraphes précédents, les composants cruciaux du système sont :

- la base de données
- les communications homme-machine
- le logiciel de base
- les opérateurs disponibles dans le (les) processeur(s).

ID3b La base de données.

Une banque de données est un ensemble formé d'une base de données, d'un système d'exploitation, de programmes d'utilisation. La base de données est un ensemble de fichiers liés logiquement caractérisé par les supports d'information, les relations entre supports (hiérarchie), la structure des liaisons entre fichiers ou éléments de fichiers.

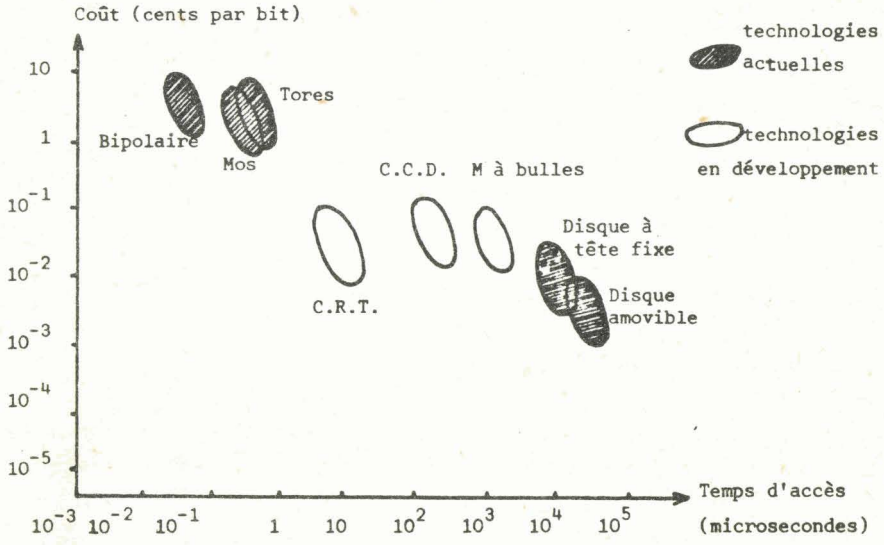
Les supports.

Certaines estimations [BRO69] définissent la base de données comme un ensemble de fichiers

- d'une capacité de l'ordre du milliard de bits
- organisé en blocs de quelques milliers de bits
- dont le temps d'accès à un bloc devrait être de l'ordre de la centaine de micro-secondes
- dont le débit devrait atteindre le million de bits par seconde.

Pour des raisons économiques et pratiques les données sont habituellement rangées sur des supports : les mémoires de masse (disque fixe ou amovible, tambour, bande). Celles-ci se caractérisent par un coût faible au bit stocké , par un temps d'accès relativement élevé, par une capacité importante [ROG74].

De nouveaux supports [WEN75] peuvent apparaître sur le marché, soit par amélioration technologique (augmentation de la densité d'information stockée par unité de surface magnétique), soit par de nouveaux moyens d'accès rapide (accès optique sur enregistrement magnétique), soit par innovation technologique (mémoires optiques de temps d'accès de l'ordre de la micro-seconde pour une capacité de l'ordre de 100 millions de bits, mémoires circulantes à matériau semi-conducteur ou magnétique dont l'accès, commandé soit par une charge électrique, soit par une paroi magnétique, se fera en une centaine de microsecondes), soit par recherche de nouvelles organisations (mémoires associatives de capacité élevée)



Comparaison de quelques technologies de mémoires [WEN 75]

L'avènement de ces nouveaux supports modifiera les systèmes informatiques ; par exemple, les mémoires optiques qui combinent la grande capacité au temps d'accès réduit permettront de réduire la hiérarchie mémoire à deux niveaux (mémoire centrale et mémoire optique) ; plus rapidement, les mémoires circulantes (ou disques électroniques) devraient permettre, en plus du remplacement des disques magnétiques, des bouleversements tels que :

- l'utilisation d'un périphérique à accès rapide comme tampon afin de régulariser le traitement
- la réduction de la taille de mémoire centrale puisque celle-ci pourra être plus rapidement rechargée
- etc ...

La hiérarchie des mémoires.

Les différences de performances entre les différentes mémoires de masse, entre celles-ci et la mémoire centrale, entraînent des questions du type :

- sur quel support placer une information ?
- sur quel support se trouve une information ?
- faut-il transférer une information d'un support à un autre ? Quand ?

Répondre à ces questions, c'est établir une hiérarchie des mémoires afin d'obtenir le meilleur compromis possible entre les performances désirées et le coût élevé d'une mémoire unique de capacité égale à celle des mémoires de masse et de vitesse équivalente à celle de la mémoire centrale.

L'exemple suivant décrit l'utilisation d'une hiérarchie de mémoires dans la gestion de transactions. [ALE74]

Soit un système de transactions bancaires où des sommes sont transférées entre comptes ; 3,5 millions de clients sont connus du système.

Celui-ci doit traiter 220 transactions par seconde en période de pointe. L'information stockée se subdivise en 3 classes :

- comptes
- tables
- programmes.

Chaque classe se subdivise elle-même en sous-classes de fréquences d'accès différentes :

Classe	Volume	(*)	sous-classe
comptes	6 M octets	45 %	a_1
	64 M octets	8 %	a_2
	640 M octets	47 %	a_3
tables	100 K octets	90 %	t_1
	9,9 M octets	10 %	t_2
programmes	1 K octets	60 %	p_1
	9 K octets	35 %	p_2
	90 K octets	4,9 %	p_3
	900 K octets	0,1 %	p_4

(*) Pourcentage d'accès d'une classe dans une sous-classe.

Une transaction consiste typiquement en :

- la consultation et modification de 2 comptes (4 accès)
- la consultation de 10 tables (10 accès)
- l'exécution de 10 000 instructions (20 000 accès)

La densité d'accès dans les différentes classes et sous-classes en période de pointe est donc (en accès M octet.$^{-1}$s^{-1}).

p_1	2,6 10^9
p_2	1,8 10^8
p_3	2,4 10^6
t_1	2 10^4
p_4	4,9 10^3
a_1	66
t_2	22
a_2	1,1
a_3	0,65

Ce type de tableau permet de choisir la localisation des différentes sous-classes dans la hiérarchie mémoire, voire de choisir les composants de cette hiérarchie.

La structure.

La structure des fichiers d'une base de données dépend en grande partie des modes de rangement et d'accès à l'information, donc du support. Les structures les plus courantes sont le chaînage, la liste, la liste multiple, la liste réciproque, le modèle relationnel, l'arborescence, etc...

ID3c Les communications homme-machine.

Elles ont lieu d'une part lors de l'étude d'une application (analyse, programmation), d'autre part lors de l'exécution des programmes correspondants.

Etude d'une application.

Deux cas peuvent se présenter ; dans le premier, le problème est étudié et résolu directement par un utilisateur, dans le second l'étude et la mise en oeuvre sont distinctes de l'exécution.

◊ Le premier cas représente généralement des opérations simples, spécifiques, programmées par des non-spécialistes. Pour cette dernière caractéristique, il est nécessaire de disposer d'un langage simple, proche de celui des utilisateurs (i.e. dont la forme soit celle d'une langue naturelle et dont le vocabulaire puisse comporter des termes techniques ayant trait à leur activité). Ce type de langage permettra par exemple des opérations telles que :

"lister le fichier XXX en le classant selon le critère YYY"

ou encore

"lister les articles du fichier WWW satisfaisant la condition C"

Des exemples de tels langages se trouvent sur des systèmes tels que SYMBAD [SYM] et REALITE 20 [INT].

◊ Dans le second cas, il s'agit d'applications interactives, répétitives, complexes et donc desquelles de bonnes performances d'exécution sont nécessaires. Leur étude est le fait de spécialistes, leur mise en oeuvre se fait par des informaticiens ; l'exécution par les utilisateurs ne laisse à ceux-ci que la possibilité de se conformer à un schéma imposé. Ces applications sont décomposées lors de l'analyse en unités de traitement simples, réalisables chacune rapidement par une ou deux personnes. Après avoir été testées individuellement, afin d'obtenir une transaction exécutable ces unités de traitement sont réunies grâce à l'utilisation d'interfaces logiciels et d'outils descriptifs permettant de définir les données et les objets agissant sur celles-ci. Un langage essentiellement déclaratif doit donc être disponible dans ce but [GRO73] [JAC75].

Exploitation.[BOR76]

L'exploitation de la gestion de transactions se caractérise d'une part par la saisie de l'information, d'autre part par les niveaux d'exploitation informatique.

La saisie.

Elle tend à être réalisée par un circuit court où le plus souvent elle est exécutée par ceux qui dans l'entreprise sont à l'origine de l'information. Ceci permet la suppression du support papier (bordereaux de perforation, cartes), grâce à l'utilisation de claviers écrans par exemple. Il faut noter que ceci accentue le besoin de sécurité car l'absence de supports papier rend difficile la reconstitution de fichiers détruits ou altérés.

Les niveaux d'exploitation.

La saisie par un circuit court est décentralisée et réalisée à tous les niveaux de qualification (magasiniers, comptables, cadres ...) et de responsabilité ; de même, la consultation et l'utilisation des données passent par un circuit court ; la décentralisation est peut être moins étendue (encore que les utilisateurs sont généralement les fournisseurs d'information) mais ici encore le personnel est non informaticien. Ceci entraîne la nécessité de réaliser simplement, sous forme de dialogues la totalité des opérations d'exploitation des applications ainsi que certains travaux liés au système d'exploitation (sauvegarde et restauration de fichiers par exemple).

ID3d Le logiciel de base.

Les caractéristiques générales d'un S.C.C. définies précédemment doivent être satisfaites par le logiciel de base : adaptativité, disponibilité, expansibilité, programmation.

D'autre part, comme pour les systèmes de commande et de contrôle, le logiciel de base doit être performant, c'est-à-dire jouer son rôle en étant le moins gourmand possible en ressources (matérielles, logicielles, temps). En effet, les transactions représentent individuellement peu d'instructions à exécuter, des entrées-sorties nombreuses mais de faible volume,

d'où le danger d'obtenir un rapport

$$\frac{\text{utilisation des ressources par l'usager}}{\text{utilisation des ressources par le système d'exploitation}}$$

beaucoup plus défavorable qu'en traitement par lots ou même qu'en temps partagé à usage scientifique [GRO73].
(Sur des machines peu adaptées à la G.T., il a été mesuré une activité du système occupant de 50 à 80 % du temps de l'unité centrale).

Enfin, le système de G.T. devant permettre à plusieurs utilisateurs simultanés de consulter et modifier la base de données, le logiciel de base doit gérer le partage de ces ressources tout en assurant un haut degré de protection. Des méthodes telles que les "C-lists" ou les "capabilities" peuvent être mises en oeuvre dans ce but [POP74] [DEN66].

ID3e Unité centrale et opérateurs.

Comme dans tout système de gestion, les opérations de manipulation de données (chaînes de caractères en particulier) sont cruciales ; en revanche les calculs arithmétiques complexes sont rares. En plus des opérations de base (chargements, stockages, opérations logiques, arithmétiques simples), il faut donner à l'unité centrale des possibilités étendues telles que :

- la recherche de sous-chaînes dans une chaîne de caractères selon divers critères (comparaison, recherche de caractères de contrôle, arrêt sur un compteur, sur une adresse ...)
- la modification de chaînes de caractères (décalages, insertions, suppressions, concaténations, expansions...)
- le tri et la fusion d'ensembles de données
- la sélection parmi un ensemble de données (recherche en tables ...)
- etc....

Certains langages évolués possèdent des opérations de ce type (LENGTH, INDEX, TRANSLATE de PL/1) ; il s'agit d'en enrichir le répertoire, d'en systématiser l'utilisation et d'en favoriser l'implantation (par exemple par le choix des micro-instructions adaptées et l'écriture de micro-programmes nécessaires).

Dans le même ordre d'idées, l'utilisation de mémoires associatives peut faciliter des opérations complexes telles que la recherche d'un ensemble de données ayant une caractéristique commune, la mise à jour d'un ensemble de données, le tri et l'inversion de fichiers, etc...

ID3f Typologie.

Deux types principaux de transactions sont utilisés :

0 la saisie purement conversationnelle, sans mise à jour immédiate de la base de données.

Dans ce cas, il ne s'agit que d'une saisie assistée qui peut mettre en oeuvre soit des contrôles internes aux données saisies (format, structure, clés, etc...), soit des contrôles internes et externes (cohérence par rapport aux données stockées par le système).

La mise à jour de la base de données se fait alors en différé (i.e. après que l'utilisateur ait considéré la transaction comme terminée) soit :

- le plus tôt possible en fonction de la disponibilité du système
- dès que le nombre de mises à jour atteint un certain seuil détecté automatiquement par le système.
- par traitement par lots après intervention humaine.

La consultation et l'utilisation de la base de données sont alors délicates puisque le résultat fourni à l'utilisateur date de la dernière mise à jour effectuée. Ce type de transaction n'est donc utilisable que

comme pis-aller ou lorsque les consultations ne requièrent pas une actualisation rigoureuse.

0 La saisie conversationnelle accompagnée d'une mise à jour en temps réel. Les contrôles sont internes et externes et la mise à jour est immédiate. La consultation fournit alors une information précise.

Ce type de transaction présente toutefois une difficulté : la performance. En effet, lorsque l'application est complexe, les mises à jour de données stockées peuvent être nombreuses et donc la transaction a une durée non négligeable, d'où une attente et une insatisfaction de l'utilisateur. Dans la pratique, on recherche souvent une solution de compromis, la transaction étant scindée en deux parties :

- la transaction proprement dite, avec saisie des données et mise à jour d'un (ou plusieurs) fichier(s) principal (aux) dont la consultation doit donner une information précise et actuelle
- un traitement en différé réalisant la mise à jour des fichiers mineurs (consultation moins exigeante en précision des informations), le plus souvent en traitement par lots en heures creuses.

Une des difficultés de l'analyse de ce type d'applications est donc d'assurer une répartition judicieuse des traitements en temps réel et en temps différé sauvegardant

- la rapidité de réponse
- les besoins des utilisateurs
- les nécessités de l'exploitation.

Exemple [GRO73]

Soit un point de vente acceptant une carte de crédit comme moyen de paiement. Il dispose d'un terminal capable de lire une carte de crédit ainsi que les commandes et données (montant de facture par exemple), et de présenter le résultat élaboré par le système (écran, imprimante, ...)

Lors de la facturation sont introduits dans le terminal

- un mot de passe et un numéro de point de vente
- la carte de crédit du client (lue directement ou recopiée grâce à un clavier)
- les caractéristiques de la vente (montant...).

Les opérations effectuées ensuite par le système seront :

1) vérification du mot de passe et de l'identité du terminal afin de savoir si le vendeur est accrédité auprès du système

2) vérification de la carte de crédit du client (validité)

3) vérification du crédit du client ou des facilités bancaires consenties (afin de déterminer si la vente est possible)

4) débit du compte du client

5) crédit du compte du vendeur

6) envoi d'un message décrivant le résultat de la transaction.

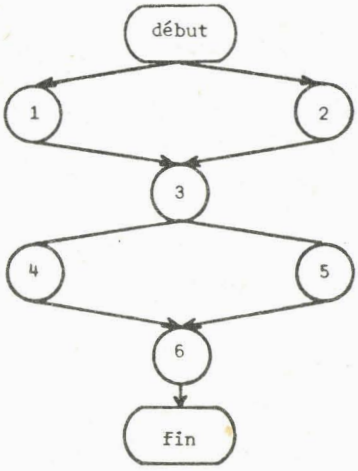
Cet exemple est simple (dialogue réduit, procédures d'erreur non précisées) mais il permet de remarquer que :

0 sur une machine classique, l'exécution de cet ensemble de traitement est séquentielle ; il apparaît toutefois que certains traitements

a) utilisent des ressources indépendantes (par exemple 1) et 2)) tant en entrée (données) que pour le traitement (si plusieurs ressources du type processeur sont disponibles). Ils peuvent donc être simultanés.

b) utilisent des ressources communes pour des traitements différents (par exemple 3) et 4) référencent le compte du client). Dans l'exemple cité ces opérations sont dépendantes l'une de l'autre mais ce n'est pas obligatoire.

0 Un graphe des principales opérations à réaliser peut être tracé :



N.B. Ce graphe n'est pas unique car certains traitements peuvent être débutés avant que certaines conditions ne soient remplies.

Conclusions.

Divers niveaux de parallélisme sont envisageables, présentant chacun des avantages et inconvénients qui le rendent adapté à certains types de travaux. Etant donné les caractéristiques de la gestion de transactions, le parallélisme est à mettre en oeuvre entre séquences d'instructions ; chaque séquence qui permettra d'exécuter une partie d'une transaction sera appelée *tâche* Par exemple, sur le graphe précédent chaque noeud pourra définir une tâche.

Une définition plus précise de la gestion de transactions consiste donc à préciser une tâche et la façon de l'obtenir, les données manipulées par une tâche, les communications entre tâches, l'enchaînement des tâches et leurs rapports avec le système d'exploitation.

ID3g **Les tâches.**

Ces séquences d'instructions qui présentent entre elles des dépendances plus ou moins grandes, des possibilités d'exécution simultanée plus ou moins riches, peuvent être définies de deux façons principales.

a) Définition par les ressources.

Sur l'exemple précédent, certaines opérations utilisent (en plus de ressources telles que espace mémoire, processeur, temps) des ressources différentes :

1) enregistrement du mot de passe et de l'identité du terminal

2) enregistrement de la carte de crédit

3) enregistrement du compte client (en lecture)

4) enregistrement du compte client (en écriture)

5) enregistrement du compte du vendeur (en écriture)

L'indépendance entre ces ressources permet l'activation des opérations correspondantes de façon indépendante. D'autre part, pour une bonne efficacité et une grande souplesse (critère de disponibilité du système), il est préférable d'allouer les ressources dynamiquement, au fur et à mesure des besoins, plutôt que de toutes les allouer en début de transaction. Ceci conduit à caractériser une tâche comme un sous-ensemble d'une transaction nécessitant un certain ensemble de ressources invariant durant son exécution.

b) Définition logique.

Dans une autre optique, la tâche peut être une entité logique, c'est-à-dire un sous-ensemble réalisant une fonction ; l'ensemble de ces fonctions compose la transaction. Sur l'exemple précédent, cette définition conduit à définir chaque noeud comme une tâche. Cette définition permet aisément l'utilisation d'une même tâche dans des applications distinctes.

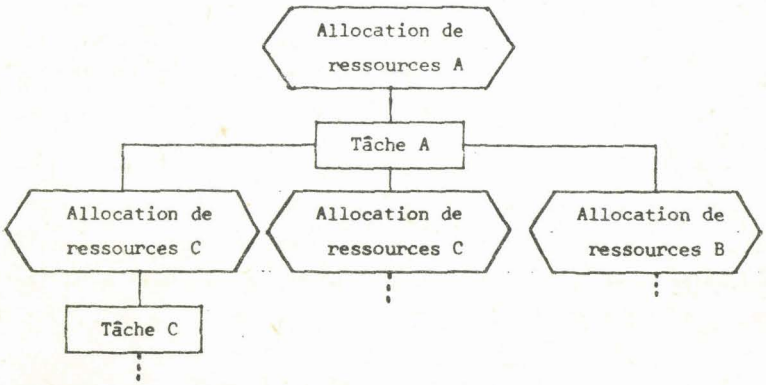
Dans bien des cas, les deux types de définitions aboutissent à la même décomposition, ce n'est pas toujours vrai et il faut choisir entre les définitions.

La première définition semble plus intéressante étant donné les contraintes de disponibilité et de partage de certains types de ressources (fichiers).

Une tâche est définie :

- par une procédure de base : séquence d'instructions spécifiant les traitements à effectuer. La ré-entrance du code est utile étant donnée la multiplicité des utilisateurs
- par les zones de communication avec d'autres tâches
- par les ressources nécessaires à l'exécution (mémoire, processeurs)
- par les interfaces avec le système d'exploitation (priorité, enchaînement des tâches, etc...).

Suivant la définition d'une tâche, une transaction peut être représentée par le schéma :



Chaque chemin du schéma représente une possibilité de transaction différente puisque chaque tâche peut, selon son résultat, déterminer quelles sont les tâches suivantes.

Le *type de transaction* est donc l'ensemble de toutes les transactions possibles lorsqu'une ressource d'un type particulier est présente en excitation du système [GRO73].

L'exemple de transaction bancaire fait partie du type de transaction "Achat-Vente", d'autres types pouvant être "Virement", "Ouverture de compte", etc...

Dans cette perspective, les traitements associés à un événement sont constitués par l'enchaînement des tâches concernées, cet enchaînement s'accompagnant des transferts de contextes nécessaires à chaque tâche activée. Le système d'exploitation doit favoriser une telle structure en activant les tâches et en transmettant les contextes correspondants.

Dans certains cas, des fonctions différentes agissent sur une ressource à partager (e.g. un fichier). Selon la nature de cette ressource et son importance, on est amené soit à partager effectivement la ressource (utilisation de segments, de descripteurs, ...) soit à n'effectuer qu'une seule opération à la fois et donc à créer et gérer une file d'attente. Cette file d'attente peut être optimisée si les tâches ont une structure permettant de n'exécuter la mise en file d'attente et le retrait qu'à des points critiques.

Enfin, la fiabilité d'un S.G.T. impose de pouvoir circonscrire les erreurs, prévoir la possibilité de les corriger, éviter les interactions indésirables entre les transactions ; le découpage en tâches paraît satisfaire cette exigence.

Une tâche étant définie par les ressources, en plus de l'analyse humaine, une aide peut être fournie aisément par la machine afin de décomposer une application en tâches. Le résultat de la compilation n'a pas

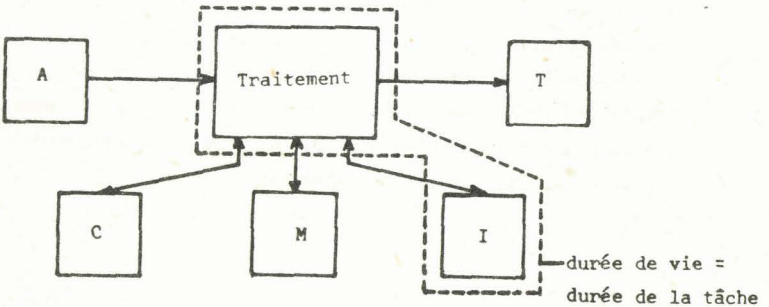
besoin d'être un "programme", mais peut être un "kit" dont les composants seront assemblés soit par une édition de liens appropriée, soit dynamiquement lors de l'exécution dans la mesure où un langage descriptif spécifie les interactions entre tâches [JAC75].

ID3h Informations d'une tâche.

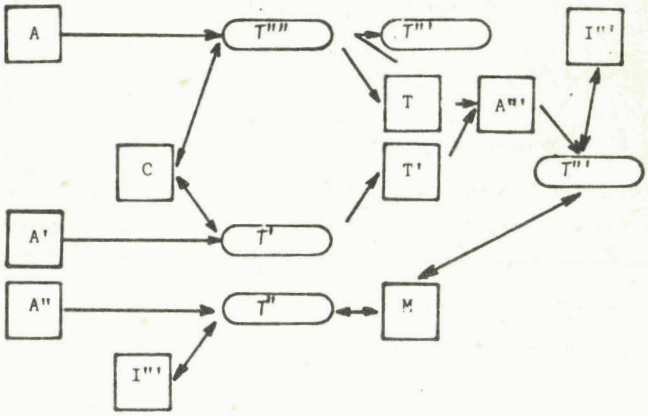
Elles peuvent se classer en :

- un contexte interne (I) ; c'est l'ensemble des informations crées par une tâche et accessibles seulement par celle-ci. Leur durée de vie est celle de la tâche.
- des contextes de communication :
  * contexte commun (C) ; c'est l'ensemble des informations utilisées par une tâche qui ne lui ont pas été transmises lors de son activation.
  * contexte argument (A) ; c'est l'ensemble des informations transmises à une tâche lors de son activation ; il disparaît en fin de tâche.
  * contexte transmis (T) ; c'est l'ensemble des informations transmises par une tâche afin d'être utilisées comme arguments par d'autres tâches.
  * contexte mixte (M) ; c'est l'ensemble des informations transmises à une tâche lors de son activation et qui seront transmises après d'éventuelles consultations et modifications.

Une tâche peut donc être schématisée :



L'ensemble des tâches actives dans le S.G.T. peut être représenté par un réseau liant leurs contextes [JAC75] :



Le contexte commun présente plusieurs inconvénients :

- certaines interactions peuvent ne pas être apparentes et donc difficiles à éliminer si elles sont indésirables
- la modification d'une partie de l'ensemble peut entraîner l'obligation de modifier la totalité du système
- la modularité des tâches n'est pas respectée.

Pour ces raisons, il est préférable d'interdire l'utilisation de contextes communs. Chaque tâche doit contenir explicitement les références des contextes nécessaires, ce qui facilite également la protection des données. Sur le schéma précédent, A"' = T ∪ T' et T = A"" ; cette situation présente certains risques et par souci de sécurité, de modularité il est préférable de distinguer les contextes arguments de chaque tâche, propriétés exclusives de la tâche à laquelle ils sont associés. Ceci entraîne des duplications de données et donc un encombrement en mémoire supérieur ainsi qu'une légère augmentation de la charge de travail.

Les contextes mixtes posent également des difficultés de sécurité et de modularité ; dans toute la mesure du possible ils sont à éviter et à remplacer par un contexte argument et un contexte transmis distincts.

Communications entre tâches.

Les communications entre tâches peuvent être réalisées par différentes méthodes :

- références à des ensembles de données permanentes (fichiers)
- boîtes aux lettres stockées soit en mémoire centrale, soit sur des supports externes (disque)
- par échanges directs entre processeurs.

Ces diverses méthodes sont à mettre en oeuvre.

a) Références à des données permanentes.

Les données permanentes forment la base de données du S.G.T. Elles permettent essentiellement les communications entre transactions. Il s'agit là de contextes mixtes qui ne peuvent être scindés comme recommandé ci-dessus.

Jackson et Moir [JAC75] les appellent des *réservoirs* (pools) et les définissent comme "des données conservées durablement" (i.e. dont la durée de vie excède celle de plusieurs transactions).

Les principales opérations effectuées sur un réservoir correspondent à des primitives

- créer
- lire
- modifier
- détruire.

Jackson et Moir présentent dans [JAC75] des exemples de mise en oeuvre. Divers types de réservoirs sont à considérer (structures différentes) et donc il y aura des familles de primitives (par exemple détruire un article d'une liste, d'un arbre, etc ...).

b) Boîtes aux lettres.

C'est le moyen essentiel de transmission des contextes arguments et transmis entre tâches.

Jackson et Moir les appellent des *canaux* (channels) et définissent un canal comme "une zone de communication utilisée afin de transmettre un message (information n'ayant d'existence et de signification que pour les deux tâches la manipulant)".

Un canal est géré par deux interfaces unidirectionnels

- produire
- consommer

Selon la structure du canal et les relations entre tâches (en particulier si elles peuvent être actives simultanément), diverses primitives d'accès sont à mettre en oeuvre.

c) Echanges directs.

Il s'agit dans ce cas de transmissions de messages entre processeurs (i.e. objets ayant en charge un processus).

Ces échanges directs sont rares entre tâches car ils posent de nombreux problèmes (en particulier sur un système multiprocesseur) parmi lesquels :

- la synchronisation

- pour la tâche émettrice des questions préalables telles que :

  l'autre tâche est-elle active ?
  sur quel processeur est-elle active ?
  comment communiquer avec elle ?
  etc ...

Ils sont plutôt réservés aux échanges entre constituants du système d'exploitation ou entre celui-ci et les transactions et les tâches.

Pour les trois types de communications décrits, une harmonisation des mécanismes d'accès et de synchronisation est souhaitable ; d'autre part ces mécanismes (notés primitives ci-dessus) sont eux-mêmes définis à partir de mécanismes de base définis et éprouvés. Les moniteurs de Hansen ou les régions critiques conditionnelles semblent particulièrement bien adaptés [HAN72] [HOAR74] [SCH76].

ID3i Enchaînement des tâches.

Etant donné les caractéristiques de la G.T., le niveau de parallélisme envisagé et la complexité des algorithmes basés sur les propriétés des graphes, l'exécution de tâches en parallèle ne doit dépendre que de la disponibilité des opérandes (et ressources) requis (contextes arguments et mixtes).

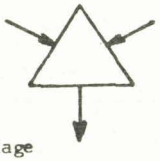
Parmi les nombreux modèles [MIL73] [BAE] [NOE73] basés sur ce principe (réseaux de Pétri, réseaux de Pétri généralisés, modèle à flux de données, ...) nous avons choisi le modèle à flux de données [RUM75] . En voici les caractéristiques.

Un programme à flux de données est un graphe d'opérateurs reliés par des liaisons de données. Un opérateur possède des entrées et des sorties de données et se caractérise par une fonction de traitement (qui pourra être complexe, par exemple la vérification d'un numéro de client) agissant sur les entrées pour fournir les sorties. Les opérateurs ne sont reliés

entre eux que par des liaisons d'entrée et de sortie ; une liaison relie une sortie d'un opérateur à une entrée d'un autre opérateur.

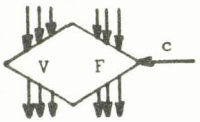
Lors de l'exécution, un opérateur utilise ses entrées, les efface et fournit les sorties. En plus des opérateurs fournis par l'utilisateur existent deux contrôleurs ; ce sont des opérateurs particuliers permettant la construction d'opérations logiques :

OU



Cet opérateur possède deux ensembles d'entrées et un ensemble de sorties ; quand une donnée apparaît sur une entrée elle est recopiée à la sortie.

Aiguillage



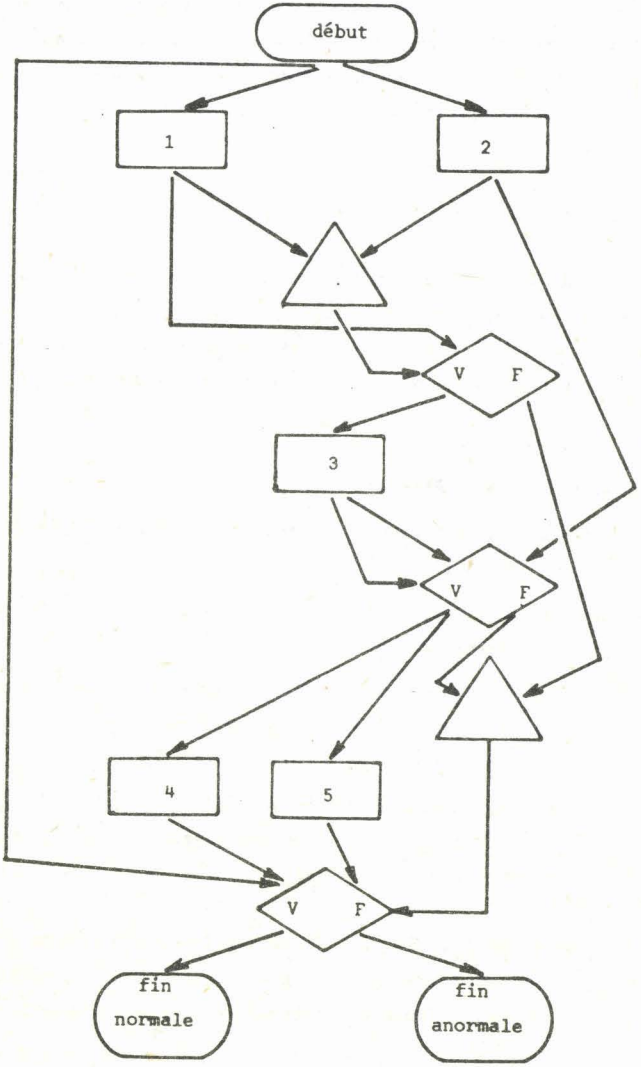
Cet opérateur possède un nombre quelconque d'entrées de types divers ainsi qu'une entrée de contrôle de type booléen. Il possède deux ensembles de sorties désignés Vrai et Faux. Lorsque toutes les entrées sont occupées, l'opérateur les recopie sur la sortie correspondant à l'état de l'entrée de contrôle (c).

Les contrôleurs permettent de mettre en oeuvre tous les types de conditions et d'opérations logiques courantes : test conditionnel, boucle, ou exclusif ...

L'exemple de transaction défini précédemment est représenté page suivante selon ce modèle.

La syntaxe permet de poser certaines conditions sous lesquelles le comportement correct du système est garanti (déterminisme, absence d'étreinte fatale). Un programme à flux de données "bien construit" se définit récursivement comme l'une des possibilités suivantes :

- un opérateur



BUS
LILLE

- un graphe acyclique de programmes
- une condition dont les branches sont des programmes disjoints
- une boucle dont le corps est un programme.

Pour mettre en oeuvre un tel modèle, des primitives de synchronisation sont à mettre en oeuvre, si possible en les intégrant aux tâches elles-mêmes afin de limiter les appels au système d'exploitation. Les moniteurs et régions critiques conditionnelles permettent de résoudre ce problème.

IE CONCLUSION.

Le système à réaliser sera considéré comme un ensemble de ressources câblées et programmées. En se limitant aux ressources câblées, on peut constituer un découpage par unités fonctionnelles. Chaque unité est douée d'une autonomie suffisante pour mener à bien un travail qui lui est confié.

Ces unités peuvent être :

* des mémoires avec l'ensemble de leurs circuits d'accès et de contrôle
* des unités arithmétiques et logiques avec les registres nécessaires à leur fonctionnement
* des unités de contrôle et de commande des programmes capables d'obtenir et d'analyser des instructions et de surveiller leur déroulement
* des contrôleurs d'entrée-sortie
* des unités diverses comme des opérateurs spécialisés ou des mémoires associatives.
* etc...

Chaque unité peut travailler d'une manière indépendante à condition qu'elle ait reçu d'une autre les consignes ou informations suffisantes et qu'elle puisse lui rendre compte du déroulement de son travail.

Ces unités sont donc totalement désynchronisées les unes des autres. Seul le dialogue autorisé par la hiérarchie établie assure leur fonctionnement collectif.

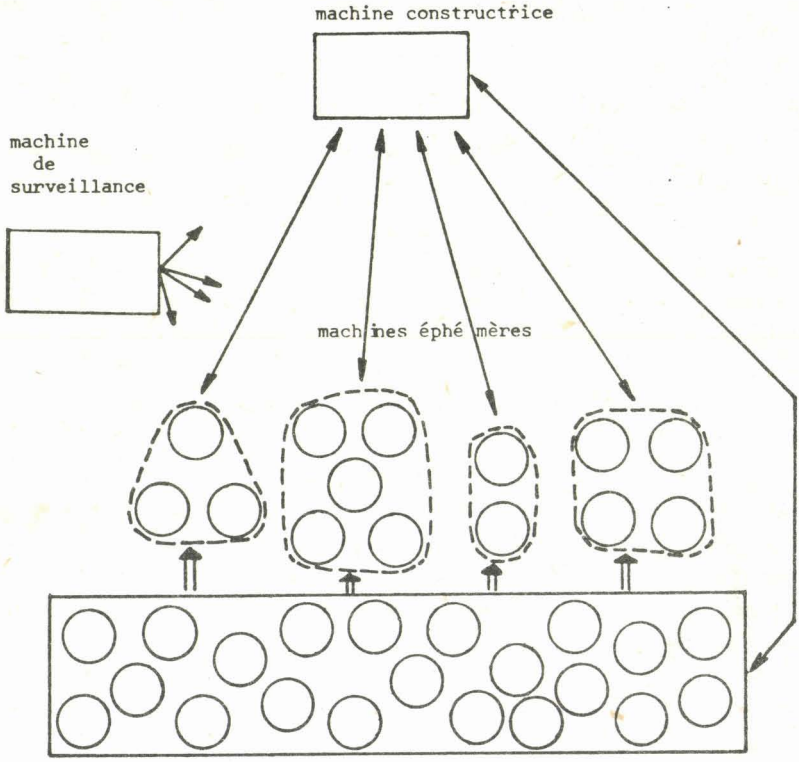
L'idée directrice est de constituer à la demande :

- pour des durées limitées au déroulement de la tâche demandée,
- avec la meilleure configuration pour cette tâche,

des machines complètes, capables de mener à bien les travaux qui leur sont confiés. Ces machines dites éphémères sont constituées par des groupements logiques d'organes tels que ceux qui ont été mentionnés plus haut ; ainsi par exemple, une machine éphémère peut regrouper deux unités de mémoire à des niveaux différents de performances, une unité arithmétique et logique, une unité de commande et trois contrôleurs de périphériques. D'où la notion d'architecture dynamique d'ordinateurs. Ces architectures doivent être capables de répondre au mieux et à tout instant à l'ensemble des demandes à satisfaire. Il faut aussi gérer les priorités entre ces demandes et reconfigurer l'ensemble afin de l'adapter à des situations nouvelles telles que pannes, saturations, défauts de ressources, apports de ressources nouvelles etc...

Un premier schéma de principe établi comporte :

- une machine constructrice gérant les algorithmes de construction des machines éphémères, tenant à jour l'inventaire des ressources disponibles et la liste des demandes de travaux
- une machine de surveillance qui teste les unités du système et détecte les pannes
- un ensemble d'unités autonomes.



Pour des raisons évidentes la machines constructrice peut être constituée à partir d'unités banalisées, du magasin de telle sorte qu'elle n'est pas irremplaçable. Il en va de même de la machine de surveillance.

Chacune des machines éphémères peut être soit un monoprocesseur, soit un groupement de processeurs travaillant sous le contrôle d'un noyau de contrôle et de synchronisation.

A la limite l'ensemble peut constituer une machine unique travaillant sur un seul problème qui exigerait toutes les ressources câblées du système.

On peut être conduit à un schéma plus fin dans lequel la fonction de construction est répartie entre plusieurs machines qui travaillent sur des parties indépendantes du magasin d'unités, qui s'échangent des ressources ou encore qui s'inscrivent dans une hiérarchie convenablement organisée.

II

LE RÉSEAU DE COMMUNICATIONS

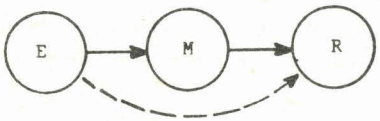
IIA INTRODUCTION.

La question étudiée dans ce chapitre est : par quel moyen technologique les objets (processeurs, mémoires, organes périphériques, etc ...) qui constituent le système vont-ils communiquer ? Etant donné l'importance d'un tel système (volume des communications par exemple) et les contraintes à respecter, un moyen simple tel que le bus paraît exclu.

IIA1 PRINCIPES DIRECTEURS [COU76]

Afin de remplir son rôle, il est souhaitable que le moyen de communication permette :

* la transparence : un transfert d'information peut être schématisé :



E = émetteur
M = moyen de communication
R = récepteur

La transparence consiste à permettre à E et R de se comporter comme en l'absence de moyen de communication (en pointillés) ; en particulier, le moyen de communication ne doit leur poser que peu de contraintes de mise en forme de l'information, de gestion de protocole, etc ...

* la modularité : ce moyen doit être composé d'éléments aussi identiques que possible afin de permettre une maintenance aisée et un coût de réalisation réduit. La modularité peut également être un élément de fiabilité en ce qu'elle permet une dégradation progressive

* l'extensibilité : pour que le système envisagé puisse être extensible, il faut que le nombre d'objets communiquant varie et donc que le moyen de communication permette une multiplication des points d'accès tout en conservant des performances convenables

* la flexibilité : la disponibilité du système exige que tout objet connecté au moyen de communication puisse échanger des informations avec tout autre objet connecté ; la reconfigurabilité est à ce prix

* l'asynchronisme : les objets connectés à ce moyen de communication sont divers par nature et par fonction, leur asynchronisme est donc probable et, vis-à-vis d'eux, le moyen de communication doit être aussi asynchrone que possible

* l'uniformité : le moyen de communication doit présenter les mêmes caractéristiques de débit, de vitesse en tous ses points d'accès et de transfert

* la simultanéité : afin d'absorber et transmettre d'importants volumes d'information, le moyen de communication doit permettre une grande simultanéité, tant dans son fonctionnement interne que vis-à-vis des objets connectés (par exemple, les transferts de A vers B et de B vers A peuvent être simultanés)

* le coût et la faisabilité : la réalisation du moyen de communication ne doit pas poser de difficulté technologique et doit se faire à un coût acceptable.

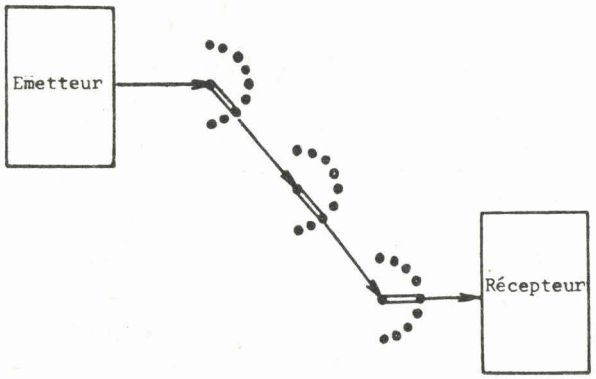
IIA2 TYPES DE COMMUTATION. [KLE76] [LEG]

La commutation est la mise en oeuvre technologique des mécanismes permettant à deux correspondants de se mettre en communication.

Deux types essentiels de commutation existent (cf. schémas)

* la commutation de lignes (ou de circuits).

Elle correspond à l'établissement d'une liaison physique entre deux correspondants. Cette liaison est spatiale et ne supporte qu'une seule communication durant son existence, ou temporelle et plusieurs communications se la partagent à tour de rôle durant des intervalles de longueur fixée. Ce type de commutation peut demander un temps d'établissement variable et non négligeable ; il est peu efficace car non partagé (sauf si la commutation est temporelle) mais permet de bonnes performances une fois la liaison établie.



Emploi d'un réseau de commutation de lignes

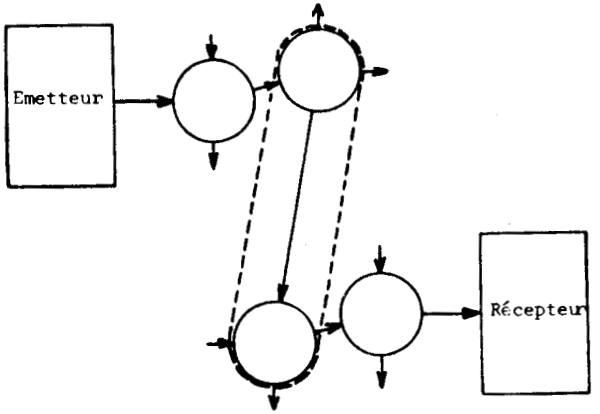
* la commutation de messages.

Elle correspond à l'établissement d'une liaison logique entre deux correspondants.

Cette liaison s'établit en 3 étapes :

- dépôt de l'information dans le moyen de communication par l'émetteur
- traitement (transfert) par le moyen de communication. Lorsque le moyen de communication est décomposable, l'analyse du transfert montre qu'au dernier niveau on aboutit à une commutation de circuits mais celle-ci ne concerne que deux éléments voisins du moyen de communication
- délivrance au récepteur.

Il y a donc stockage par le support ; un cas particulier est donné par les systèmes "store-and-forward" où l'information est stockée puis transmise par une liaison élémentaire, puis re-stockée, etc...



Emploi d'un réseau de commutation de messages.

Ce type de commutation permet généralement un meilleur partage du moyen de communication et donc une meilleure simultanéité. Un cas particulier très important est la commutation de paquets [KLE76].

IIA3 QUELQUES MOYENS EXISTANTS.

Citons : les liaisons par bus (simples, hiérarchisés, en temps partagé), les liaisons par boîtes aux lettres, les liaisons matricielles ("cross-bar switch", "multiport"), les liaisons par "noria" (à une ou plusieurs dimensions), les liaisons par réseau à décalage (ILLIAC IV), les réseaux, etc...

Cousin [COU76] a montré les avantages d'une structure de réseau, non seulement en ce qu'elle satisfait aux principes précédents, mais également de par ses performances (débit élevé, vitesse de propagation, etc...) ; nous avons adopté ses conclusions.

IIA4 STRUCTURES DE RESEAUX.

Un réseau se définit comme un ensemble d'éléments modulaires communiquant entre eux.

Un réseau cellulaire [WAG66], [COU76] est formé d'un ensemble de cellules identiques reliées entre elles et avec des objets connectés ; il met en oeuvre la commutation de messages par stockages et transferts successifs ("store-and-forward").

Afin de jouer son rôle, une cellule est :

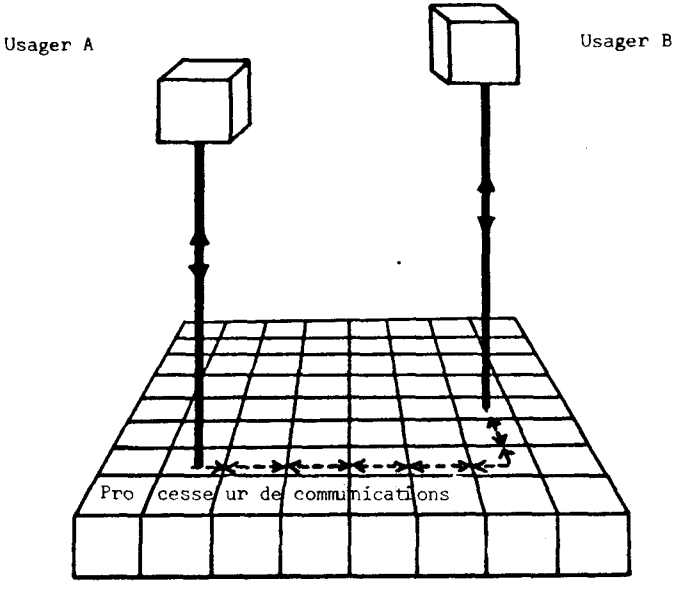
- un point d'entrée vers le réseau de l'information issue d'un objet connecté
- un point de mémorisation de l'information en cours d'acheminement
- un point de transfert d'information qu'elle contient vers une cellule voisine (i.e. qui lui est directement connectée)

◊ un point de sortie du réseau vers un objet connecté.

Le graphe des connexions d'une cellule, fonction des contraintes sur le moyen de communication, est caractérisé par le nombre p d'arêtes issues d'un sommet ou cellule. Le choix de Cousin (p=4) l'a amené à définir la structure du P.C.C.

IIB DESCRIPTION DU PROCESSEUR CELLULAIRE DE COMMUNICATIONS. [COU76]

Le Processeur Cellulaire de Communications (P.C.C.) étudié par Cousin, dont cette section IIB rappelle les conclusions, est un réseau plan carré, de structure matricielle (cf. schéma) ; si l'arête est composée de n cellules, le réseau en comporte n^2 toutes identiques par construction. C'est un réseau ouvert, c'est-à-dire que les cellules situées aux deux extrémités d'une même rangée (ligne ou colonne) ne sont pas reliées directement par un (ou plusieurs) arc(s).



IIB1 DESCRIPTION D'UNE CELLULE GRAPHE DE CONNEXIONS.

Une cellule se compose :

- d'une logique câblée de contrôle dont le rôle sera précisé en IIB2
- de six registres de stockage de l'information à transférer
- de liaisons vers les quatre cellules voisines ainsi que vers l'élément de traitement, de contrôle, de mémoire .... qui lui fait face.

Sur le schéma de la page suivante sont indiqués les registres nommés Nord, Est, Sud, Ouest, Récepteur, Emetteur (notés respectivement N, E, S, O, R, M) ainsi que les liaisons permises entre registres d'une même cellule ou de cellules voisines. Ce schéma représente un réseau 3×3 ; c'est le plus petit réseau où tous les types de cellules (identiques par construction mais différentes de par les registres et liaisons utilisés selon leur position) sont présents. La relation "la cellule x utilise les mêmes arcs que la cellule y", notée x R y, est une relation d'équivalence qui permet de définir neuf classes d'équivalence. Dans un réseau n×n où une cellule est repérée par son indice de ligne et son indice de colonne ces classes sont :

$$C_I = \{(1,1)\}$$

$$C_{II} = \{(1,j) \mid j = 2...n-1\}$$

$$C_{III} = \{(i,n)\}$$

$$C_{IV} = \{(i,1) \mid i = 2...n-1\}$$

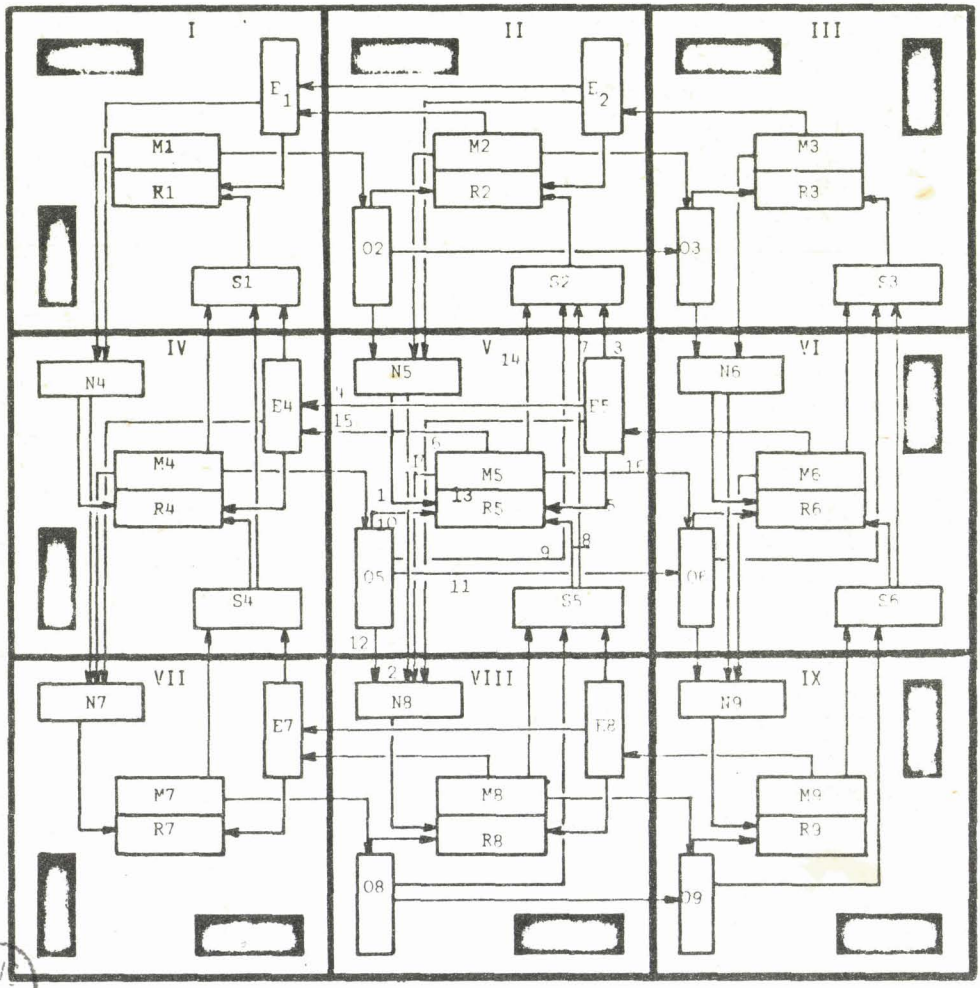
$$C_V = \{(i,j) \mid i = 2...n-1,\ j = 2...n-1\}$$

$$C_{VI} = \{(i,n) \mid i = 2...n-1\}$$

$$C_{VII} = \{(n,1)\}$$

$$C_{VIII} = \{(n,j) \mid j = 2...n-1\}$$

$$C_{IX} = \{(n,n)\}$$



N.B. Les registres noircis sont inutilisés ; les liaisons non utilisées ne sont pas représentées.

Sur la cellule centrale du schéma (3,3) sont présents tous les arcs partant et aboutissant à une cellule. Il y a dix-sept arcs dont seize sont internes au réseau, le dernier (numéroté treize) représentant le composant connecté à la cellule. Le tableau suivant reprend, pour chaque classe, les arcs utilisés.

Classe	Arcs
I	5, 6, 8, 13, 16, 17
II	4, 5, 6, 8, 10, 11, 12, 13, 15, 16, 17
III	8, 10, 12, 13, 15, 17
IV	1, 2, 3, 5, 6, 7, 8, 13, 14, 16, 17
V	1 à 17
VI	1, 2, 7, 8, 9, 10, 12, 13,14, 15, 17
VII	1, 3, 5, 13, 14, 16
VIII	1, 3, 4, 5, 9, 10, 11, 13, 14, 15, 16
IX	1, 9, 10, 13, 14, 15

Etant donné l'ensemble des liaisons vues précédemment, la matrice caractéristique du graphe (dont les sommets sont les registres) représentant le réseau peut être calculée.
Le réseau 3×3 comprenant tous les types de cellules, cette matrice caractéristique permet le calcul de la matrice caractéristique d'un réseau de dimension quelconque.

Grandeurs caractéristiques du graphe ($n \geq 3$)

* Nombre de cellules : n^2
* Nombre de registres : $6n^2$
* Nombre de registres utilisés : $6n^2 - 4n$

* Nombre d'arcs utiles internes au réseau :

$$\underbrace{5 \times 4}_{\text{classes I,III,VII,IX}} + \underbrace{10 \times 4 \times (n-2)}_{\text{classes II,IV,VI,VIII}} + \underbrace{[n^2 - 4 - 4(n-2)] \times 16}_{\text{classe V}}$$

Soit au total $\underline{16n^2 - 24n + 4}$ arcs internes utiles

et donc $\underline{\underline{17n^2 - 24n + 4}}$ arcs utiles.

n	n^2	$6n^2 - 4n$	arcs utiles internes	arcs utiles	nombre moyen d'arcs utiles par cellule
3	9	42	76	85	9,44
4	16	80	164	180	11,25
5	25	130	284	309	12,35
6	36	192	436	472	13,11
7	49	266	620	669	13,65
8	64	352	836	900	14,05
9	81	450	1084	1165	14,38
10	100	650	1364	1464	14,64
11	121	682	1676	1797	14,85
12	144	816	2020	2164	15,02
13	169	962	2396	2565	15,18
14	196	1120	2804	3000	15,31
15	225	1290	3244	3469	15,42

Le tableau ci-dessus présentant les valeurs numériques caractéristiques d'un réseau de dimension n montre que le nombre très élevé d'arcs permet un grand nombre de transferts simultanés ; d'autre part le nombre moyen d'arcs utiles par cellule tend rapidement vers sa valeur maximale d'où une bonne utilisation du matériel investi.

IIB2 CHRONOGRAMME.

Le transit d'une information à travers le P.C.C. est un processus évoluant dans le temps et dans l'espace. Ce processus naît lors de l'injection de l'information dans le réseau et meurt lors de son extraction. Le P.C.C. présente un haut degré de parallélisme ; à un instant donné il y a donc présence d'une famille de processus P_{ijk} dans le réseau (i,j : indices de cellule, k : indice du registre contenant l'information à transférer). Pour évoluer, un processus doit disposer des ressources suivantes :

- procédures câblées d'acheminement
- informations sur le trajet à suivre, contenues dans l'information à transférer (i.e. dans le registre k)
- un registre destinataire de l'information.

Les procédures câblées d'acheminement (test d'état, demande de transfert...) sont attachées au registre k et sont donc acquises au processus P_{ijk}. En revanche, le registre destinataire peut être une ressource commune et critique à plusieurs processus. L'exclusion mutuelle entre processus demandeurs doit donc être assurée. A cet effet, le processus P_{ijk} est décomposé en deux processus P^1_{ijk} et $P^2_{mn\ell}$. Le processus P^1_{ijk}, attaché au registre k de la cellule (i,j) est chargé d'analyser le contenu du registre k et de demander le transfert de l'information vers un registre C^{ℓ}_{mn}.

Le processus $P^2_{mn\ell}$, attaché au registre C^{ℓ}_{mn} est chargé d'analyser les demandes, d'en sélectionner une et d'exécuter le transfert ; il réalise donc l'exclusion mutuelle.

De cette décomposition en deux processus résulte le chronogramme établi par Cousin décrivant un cycle de base du réseau.

Processus P^1_{ijk}	Analyse du contenu du registre k	Activation des $P^2_{mn\ell}$			
Processus $P^2_{mn\ell}$			Analyse des demandes	Sélection d'un P^1_{ijk}	Transfert

IIB3 DESCRIPTION DU CONTENU D'UN REGISTRE.

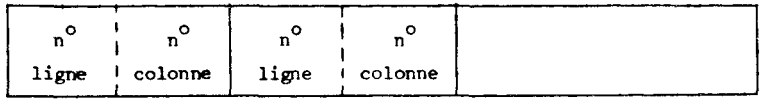
Afin de mettre en oeuvre le mécanisme précédent, le registre C^k_{ij} doit contenir

- l'information utile à transférer
- des informations "de service" nécessaires aux processus P^1_{ijk} et $P^2_{mn\ell}$ (spécification de la destination).

Une spécification d'origine de l'information à transférer peut également être utile au fonctionnement du composant destinataire. Le contenu d'un registre, appelé message, comprend donc :

- la spécification d'origine
- la spécification de destination
- l'information destinée au composant destinataire
- d'éventuelles informations de contrôle (parité par exemple).

L'adressage est réalisé sous forme matricielle, une cellule étant définie par le couple (n° de ligne, n° de colonne) ; cette notation ne présente aucune ambiguïté et se prête bien au traitement par des circuits comparateurs.



Format d'un message.

Si l'on se limite à un réseau 8×8, les adresses nécessitent 6 bits chacune ; l'information à transférer est par exemple de 16 bits. Le message occupe donc environ 32 bits ; ce chiffre attire deux remarques :

- le taux $\frac{\text{information transférée}}{\text{format d'un message}}$ est de l'ordre de 1/2, ce qui est peu et indique que le rapport performance/investissement n'est pas aussi favorable qu'espéré

- ce taux peut être amélioré si l'on augmente la quantité d'information "utile" d'un message. Nous nous heurtons alors à un problème technologique car le nombre de connexions devient prohibitif (de l'ordre de deux-cent connexions par cellule pour un message de trente-deux bits).

IIB4 MODE DE ROUTAGE.

Les problèmes d'interblocage par présence de circuits ont amené Cousin à choisir un réseau plan ouvert où le routage s'effectue de la manière suivante : "tout transfert d'un message inséré dans la cellule C_{ij} vers la cellule $C_{k\ell}$ (où il est extrait) est une succession de transferts élémentaires d'abord horizontaux (de C_{ij} à $C_{i\ell}$), puis verticaux (de $C_{i\ell}$ à $C_{k\ell}$)". Un tel routage permet de n'utiliser que les liaisons déjà définies en évitant tout risque d'interblocage.

IIC ÉTUDE DE QUELQUES ASPECTS COMPLÉMENTAIRES DU P.C.C.

Cette partie concerne l'étude de quelques aspects du P.C.C. tendant soit à en définir la sensibilité à des phénomènes tels que la localité des communications, l'hétérogénéité des éléments connectés, soit à en préciser le comportement en introduisant un interface en tampon entre une cellule et les objets qui lui sont connectés ou en lui donnant une vitesse supérieure à celle des objets connectés. Dans ce but ont été effectuées de nouvelles simulations décrites en annexes avec leurs résultats essentiels (Annexe I).

IIC1 PRINCIPE DE LOCALITE.

Principe.

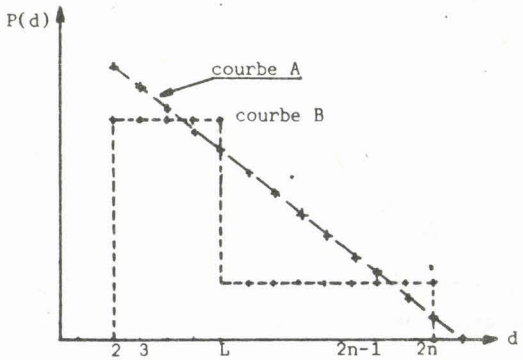
Dans les simulations qu'il a effectuées, Cousin a supposé qu'un objet quelconque du système peut communiquer avec tout autre objet connecté selon une probabilité constante, indépendante des positions relatives des objets considérés.

C'est là se placer dans les conditions les plus défavorables car il semble que si les communications se font de préférence dans le proche voisinage, la charge du réseau est réduite et donc les performances (vitesse, capacité) sont améliorées.

La contrainte est alors de favoriser ces communications locales par une politique d'allocation adéquate. Afin de justifier cette charge supplémentaire du système d'exploitation, des simulations ont été effectuées pour évaluer l'amélioration obtenue.

Le principe de localité mis en oeuvre a pris deux formes :

- une localité linéaire selon la courbe A
- une localité en escalier selon la courbe B



Probabilité de communication en fonction de la distance.

Résultats de simulation.

La première loi a donné des résultats médiocres, par contre l'amélioration obtenue dans le second cas justifie une politique d'allocation tenant compte de la localité (si elle n'est pas trop gourmande en ressources).

Les résultats essentiels sont que, avec localité "en escalier" :

- le temps moyen de parcours d'un message dans le P.C.C. de dimension n tend vers le temps moyen de parcours dans le P.C.C. de dimension L (L, distance de localité définie sur le schéma précédent)

- le nombre moyen d'injections dans un P.C.C. de dimension n est pratiquement égal au nombre moyen d'injections dans un P.C.C. de dimension L multiplié par le rapport du nombre de cellules.

$$\bar{I}_k(n) = \frac{n^2}{L^2} \bar{I}_k(L)$$ pour un même taux d'émission.

IIC2 HETEROGENEITE DES OBJETS CONNECTES.

Principe.

Selon leur nature, voire selon le travail qui leur est confié, les objets connectés au P.C.C. ont des comportements différents. En particulier, le taux d'émission est variable selon le type d'objet ; le P.C.C. est très sollicité en certains points alors qu'en d'autres la charge est faible.

Résultats de simulation.

3 types d'objets étaient simulés dans un réseau 8*8 selon diverses configurations parmi lesquelles :

3	1	3	3	3	1	3	3
2	3	3	1	3	3	2	1
3	3	2	3	2	3	3	3
1	3	3	3	3	1	2	3
3	2	1	3	3	3	3	1
3	3	3	2	3	2	3	3
1	2	3	3	1	3	3	2
3	3	1	3	3	3	1	3

A
(Localité 2)

3	1	3	3	3	1	3	3
3	3	3	1	3	3	3	1
1	3	3	1	3	3	3	3
3	3	3	3	3	1	1	3
3	1	1	3	3	3	3	3
3	3	3	3	1	3	3	1
1	3	3	3	1	3	3	3
3	3	1	3	3	3	1	3

B
(Localité 2)

3	3	3	3	1	3	3	3
3	1	3	3	3	3	3	3
3	3	3	3	3	3	3	1
3	3	3	1	3	3	3	3
3	3	3	3	1	3	3	3
1	3	3	3	3	3	3	3
3	3	3	3	3	3	1	3
3	3	3	1	3	3	3	3

C
(Localité 3)

Caractéristiques des différentes configurations :

- les objets de type 1 ont un taux d'émission de l'ordre de 0.4
- les objets de type 2 ont un taux de l'ordre de 0.05
- les objets de type 3 ont un taux de l'ordre de 0.1.

Ce sont par exemple respectivement des mémoires, des unités d'entrée-sortie, des processeurs.

Si l'on calcule le taux moyen sur les différents réseaux, il vient :

A : $\bar{\lambda} = 0.148$ (42 objets de type 3, 12 de type 1, 10 de type 2)

B : $\bar{\lambda} = 0.175$ (48 objets de type 3 et 16 de type 1)

C : $\bar{\lambda} = 0.1375$ (56 objets de type 3 et 8 de type 1).

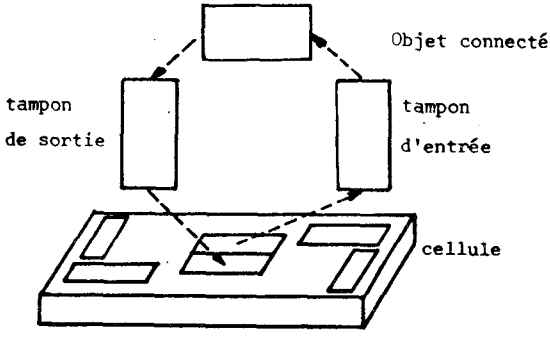
La principale conclusion des simulations est que le P.C.C. se comporte globalement comme s'il était chargé au taux $\bar{\lambda}$ uniforme. Les points de forte charge ($\lambda > \bar{\lambda}$) sont compensés par les points de faible charge ($\lambda < \bar{\lambda}$).

IIC3 INTERFACE TAMPON.

Justification.

Cousin a montré le danger d'un rapide blocage du réseau par suite de panne d'une cellule ; lorsqu'un objet connecté tombe en panne et refuse de retirer tout message du P.C.C., le danger est le même. Certains objets connectés (processeurs par exemple) peuvent être dans l'impossibilité de retirer des messages bien que fonctionnant correctement ; c'est le cas lors de l'exécution d'une routine ininterruptible par un processeur connecté. Il est donc nécessaire, pour assurer le bon fonctionnement du P.C.C., de placer entre le registre récepteur d'une cellule et le (ou les) objet(s) connecté(s) au P.C.C. un tampon de type FIFO qui "éponge" les messages et les stocke en attendant que l'objet connecté les prenne en compte. Ainsi le risque de blocage est fortement atténué, d'autant plus que le tampon est de dimension importante.

D'autre part, le P.C.C. peut subir localement une forte charge, ce qui entraîne que des registres émetteurs sont occupés durablement. L'objet connecté ne peut alors y déposer son information ce qui lui impose soit d'attendre, soit de détruire le message (s'il n'a pas de possibilité de stockage ou si ses ressources sont requises par un autre travail). Cette attente peut être supportable (par exemple si aucun travail n'attend) mais peut être pénalisante. Un tampon FIFO placé entre la sortie d'un objet connecté et le registre émetteur de la cellule permet donc une meilleure utilisation des objets connectés.



De telles piles tampon ont également l'avantage de découpler le P.C.C et l'objet connecté afin de rendre possible des vitesses de travail différentes et donc :

- des différences importantes entre les vitesses (un rapport 10 ou plus est ainsi réalisable)
- la connexion de différents types d'unités de vitesses différentes à un même P.C.C.

Enfin, il semble que ces piles tampon jouent un rôle régulateur qui permet des flots d'information plus continus et uniformes et donc de meilleurs traitements. Leur inertie (messages contenus) peut toutefois poser des problèmes en cas de reconfiguration.

Mise en oeuvre.

En plus de leur rôle de stockage, les piles composant l'interface ont à déposer l'information et à la retirer du P.C.C. Ces piles doivent donc être de petits automates ; diverses fonctions ayant trait aux communications peuvent y être implantées pour décharger les objets connectés de tâches annexes (afin d'améliorer la transparence du P.C.C. pour ceux-ci)

- modification de format
- contrôles d'erreur de routage
- génération et contrôle de parité
- etc...

Ces fonctions sont plus précisément définies dans le chapitre sur les protocoles de communication (Ch. III).

Simulations.

Les simulations ATHENES ont mis en oeuvre ces piles tampons et ont donné les résultats suivants :

0 sur un P.C.C. saturé, il y a baisse de performances tant sur le nombre moyen d'injections que sur le temps de parcours

◊ sur un P.C.C. non saturé, la durée de vie (entre création et réception) est augmentée (charge du P.C.C. accrue, temps d'attente dans les files) et très dispersée. Lorsqu'un objet réclame une information par le P.C.C., la réponse peut donc lui parvenir très longtemps (jusque 200 cycles ou plus) après. Il peut être contraint à une attente forcée (de la réponse) non négligeable (et plus importante que l'attente d'injection en l'absence d'interface)

◊ sur un P.C.C. non saturé, la probabilité d'injection dans le P.C.C. tend vers λ (taux théorique d'émission).

Conclusions.

Bien qu'utiles pour un bon fonctionnement du P.C.C. (blocages) les interfaces ont un effet néfaste sur le fonctionnement des objets (attentes) et sont donc à mettre en oeuvre avec beaucoup de prudence. On peut limiter leur rôle aux situations critiques décrites plus haut en évitant de les employer lorsque cela n'est pas nécessaire. Il n'en demeure pas moins que leur apport aux questions de sécurité de fonctionnement justifie une perte de performances.

IIC4 PROBLEMES DE SYNCHRONISATION.

Justification.

Les objets envisagés dans le système sont :

* des mémoires.

Selon leur prix , leur capacité, leur technologie, leur vitesse varie dans de grandes proportions de lµs à 100 ns environ

* des microprocesseurs.

Selon leur technologie (bipolaire ou MOS), leur architecture, selon le type d'instruction utilisée, les résultats sont disponibles (et donc transmissibles) à une cadence allant de 300 ns (INTEL 3000) à plusieurs micro-secondes (8080 INTEL, F8 Fairchild, 2650 Signetics, etc...)

* des objets spécialisés tels que processeurs de périphériques, mémoires associatives, etc ... dont les caractéristiques sont très variables.

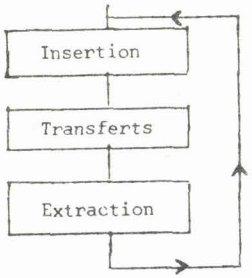
L'étude de faisabilité réalisée à la suite du travail de Cousin permettait d'espérer un cycle de base du P.C.C. de quelques centaines de ns, donc inférieur à celui de certains des objets connectés. Comme il a été précisé précédemment, il faut permettre l'implantation d'objets de caractéristiques différentes (critère d'extensibilité du système). La simulation de cette variété de cycles étant délicate, l'étude a été limitée à un ensemble d'objets de même vitesse. Le système est alors caractérisé par le rapport

$$R = \frac{\text{cycle objet}}{\text{cycle P.C.C.}} \geq 1$$

que nous avons pris entier afin de faciliter la synchronisation.

Mise en oeuvre de la simulation.

Le chronogramme utilisé par Cousin peut être modifié et ré-utilisé afin d'avoir R entier ≥1.



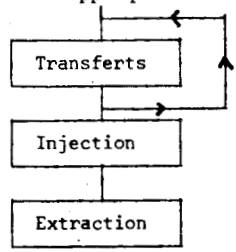
Dans ce cas, durant 1 cycle processeur il y a R cycles P.C.C. et les injections, vues du processeur au taux λ_o, sont vues du réseau au taux λ_o/R. L'étude de Cousin reste alors valable moyennant le facteur 1/R.

Généralement, deux cas se présentent :

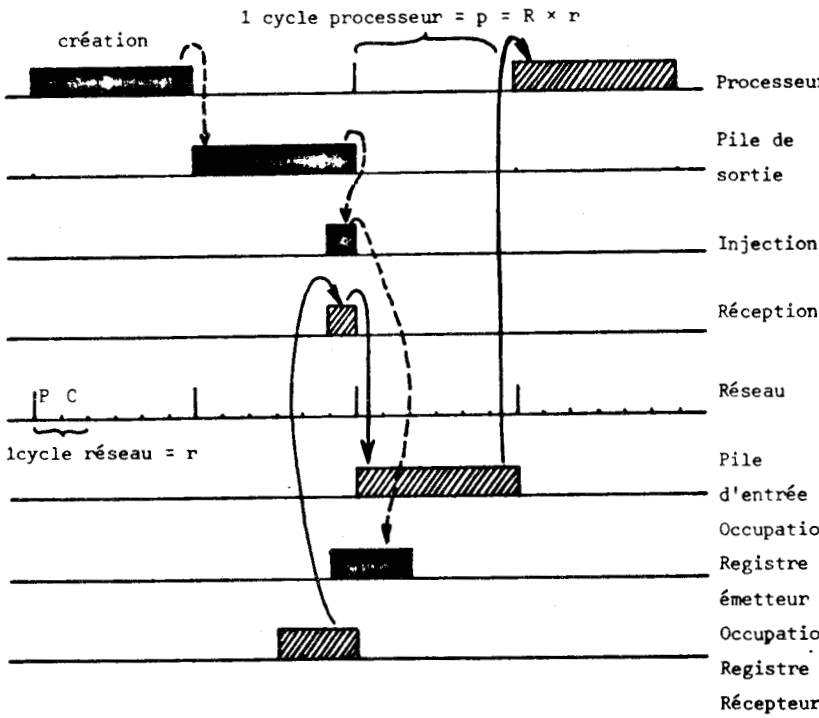
◊ il y a interface, celui-ci peut avoir même cycle de base que le P.C.C. mais peut avoir un cycle supérieur, en particulier si l'interface assume certaines fonctions annexes.

0 il n'y a pas d'interface, le cycle est donc supérieur (les objets sont généralement plus lents) et le chronogramme de Cousin est impossible..

L'organigramme suivant s'applique alors.



D'où le diagramme temporel (exemple pour R = 3) :



Remarques :

* un message crée au cycle n de l'objet (cycle nR du P.C.C.) parvient au plus tôt dans le registre émetteur au cycle réseau (n+1)R car la simulation supposait qu'un cycle objet au moins devait être passé en tampon.

* un message traité au cycle n de l'objet a été reçu au plus tard dans le registre récepteur au cycle P.C.C. (n-1)R-2 (soit au cycle objet n-2).

* le cas R=1 est traité de la même façon que par Cousin.

* les injections et extractions se font simultanément à des transferts mais sont moins prioritaires.

Résultats de simulation.

Comme c'était prévisible, plus R est grand, plus le fonctionnement du P.C.C. devient "fluide" :

- nombre d'injections tendant vers λn^2
- temps de parcours dans le P.C.C. tendant vers 1.

Le P.C.C. s'éloignant de la zone de saturation, il devient possible d'y connecter des éléments à fort taux d'émission, même sans localité.

Enfin, la distribution des temps d'attente et de parcours dans le P.C.C. étant beaucoup plus concentrée, l'utilisation de files tampon peut se faire sans difficultés lorsque R augmente.

IIC5 LIAISONS MULTI-MESSAGES.

Matrice de communications.

Les communications entre éléments connectés à un quelconque moyen de communications peuvent être caractérisées par la matrice de communications

T dont un terme t_{ij} représente le temps moyen de transfert de l'élément i à l'élément j. Les éléments diagonaux de cette matrice sont nuls ; généralement les matrices sont symétriques, i.e. $t_{ij} = t_{ji}$. Pour le P.C.C., l'adressage d'une cellule étant de type matriciel, la matrice de communications T se définit selon 4 dimensions :

$$t_{ijk\ell} = \text{temps moyen de transfert de } (i, j) \text{ vers } (k, \ell).$$

Propriétés de ce tableau n × n × n × n (n = dimension du PC.C.C.) :

◊ il reflète non seulement les propriétés du P.C.C. mais également celles de la configuration (i.e. de la répartition) des objets connectés puisqu'un terme $t_{ijk\ell}$ est fonction de la charge sur le parcours de (i, j) vers (k, ℓ).

◊ les trajets de (i, j) vers (k, ℓ) et de (k, ℓ) vers (i, j) sont différents et ont des charges différentes, donc généralement :

$$t_{ijk\ell} \neq t_{k\ell ij}$$

◊ cette matrice ne comprend aucun terme infini puisque le P.C.C. permet des communications entre des éléments de positions quelconques

◊ cette matrice peut également être définie en tenant compte de l'allocation des éléments du système à un instant donné. Dans ce cas elle comporte des termes infinis correspondant aux éléments qui ne peuvent communiquer vu l'état d'allocation (en particulier, une localité totale rend certains termes infinis).

Obtention de cette matrice :

- par simulation
- par mesures sur un réseau réel
- par calcul : connaissant les caractéristiques d'émission des objets, leur configuration et si le P.C.C. n'est saturé en aucun point, il est possible de calculer la charge de communications sur chaque arc du P.C.C..Par sommation, une matrice T' décrivant la charge du P .C.C. sur les trajets (i, j) vers (k, ℓ) est obtenue. Cette matrice T' (cf. annexeII)

décrit la difficulté (ou plus exactement le ralentissement par rapport à la situation idéale où le P.C.C. est vide) à transmettre une information de (i, j) vers (k, ℓ). Elle est bien évidemment reliée très étroitement à la matrice T par une fonction Ω faisant correspondre le temps moyen de transfert de (i, j) vers (k, ℓ) à la charge.

Liaisons multi-messages.

Un message du P.C.C. est de taille limitée ; dans de nombreux cas l'information à transmettre ne peut s'en accomoder et plusieurs messages sont nécessaires pour la transférer. Dans ce cas, l'objet émetteur envoie la séquence $m_1, m_2, \ldots, m_z$ qui est reçue dans l'ordre par l'objet récepteur mais cette séquence peut être mélangée avec d'autres séquences émises par d'autres objets vers le même récepteur. Celui-ci reçoit donc par exemple :

$$m_1, m'_1, m_2, m'_2, m_3, m_4, m'_3, m'_4, \ldots, m_z$$

où m'_1, m'_2, m'_3, m'_4 est une séquence émise par un autre objet vers le même récepteur. Avant traitement par l'objet récepteur celui-ci doit donc trier les séquences afin de reconstituer des séquences homogènes, (i.e. de même émetteur et véhiculant l'information considérée, à l'exclusion par exemple d'une seconde information de même émetteur émise par la suite et pouvant n'avoir aucun rapport avec la première). Ceci imposerait de nouvelles fonctions à l'interface objet-cellule et donc une plus grande complexité et une lenteur accrue ; les messages devraient également comporter des informations de séquences d'où une perte d'information "utile". Cette émission de plusieurs messages permettrait toutefois de mettre à profit l'aspect pipe-line du P.C.C. puisque le temps de transfert de z messages consécutifs de (i, j) vers (k, ℓ) est plus faible que z fois le temps de transfert d'un message.

Afin de tenir compte de cet aspect, la matrice de communications T comprend une dimension supplémentaire et un terme $t_{ijk\ell s}$ correspond au temps moyen de transfert de s messages consécutifs issus de (i, j) vers (k, ℓ). L'aspect pipe-line s'exprime par l'inégalité :

$$t_{ijk\ell s} \leq s \times t_{ijk\ell 1} \quad \forall s \text{ entier} > 0$$

IID MODÉLISATION DU P.C.C.

IID1 PRESENTATION ET HYPOTHESES.

Cette partie vise à modéliser le réseau, c'est-à-dire à définir son comportement grâce à une méthode de calcul. Cette modélisation n'est valide que dans certaines limites :

1) le réseau est supposé infini
2) un principe de localité "en escalier" est appliqué ; cette localité est supposée parfaite, c'est-à-dire que tous les messages ne peuvent parcourir plus d'une distance L dans le réseau
3) le modèle ne définit que les valeurs moyennes des variables caractéristiques du réseau
4) les composants connectés au P.C.C. ont tous un même comportement vis-à-vis de celui-ci.

Ces restrictions permettent d'obtenir un modèle simple donnant une bonne approximation du comportement d'un P.C.C. de dimension finie à localité imparfaite.

Avant de définir ce modèle, les métriques dans le P.C.C. sont étudiées ; ensuite le modèle est présenté ainsi que certaines généralisations.

IID2 METRIQUES DANS LE RESEAU.

IID2a Notion de distance moyenne.

Dans un réseau de n^2 cellules, la distance moyenne d'une cellule (x_o, y_o) au reste du réseau est définie par :

$$\bar{d}_{x_o y_o} = \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} d^{ij}_{x_o y_o} \, \delta^{ij}_{x_o y_o} \right\} \frac{1}{n^2 - 1}$$

où $\delta^{ij}_{x_o y_o} = 0$ si $(i, j) = (x_o, y_o)$; $= 1$ sinon. Où $d^{ij}_{x_o y_o}$ représente la distance entre deux cellules (x_o, y_o) et (i, j).

Sur le P.C.C., la distance entre deux cellules (x, y) et (x_o, y_o) est définie comme le nombre d'arcs à parcourir sur le graphe (dont les sommets sont les registres). Cette distance vaut :

$$|x - x_o| + |y - y_o| + 1$$

La distance moyenne s'écrit donc :

$$\bar{d}_{x_o y_o} = \frac{1}{n^2 - 1} \sum_{\substack{x=1 \\ x \neq x_o}}^{n} \sum_{\substack{y=1 \\ y \neq y_o}}^{n} (|x - x_o| + |y - y_o| + 1)$$

Après calcul, il vient :

$$\bar{d}_{x_o y_o} = \frac{n^3 + 2n^2 - n x_o - n y_o + n x_o^2 + n y_o^2 - n^2 x_o - n^2 y_o - 1}{n^2 - 1}$$

Sur un P.C.C. d'arête n, la distance moyenne sur tout le réseau se définit par :

$$\bar{d}_n = \frac{1}{n^2} \sum_{x_o=1}^{n} \sum_{y_o=1}^{n} \bar{d}_{x_o y_o}$$

D'où $\bar{d}_n = \frac{2n}{3} + 1$ (réseau plan ouvert) $\forall n$ entier > 1.

La distance moyenne ainsi définie est caractéristique du P.C.C. quels que soient les objets qui lui sont connectés ; la définition suivante va elle tenir compte du comportement de ces objets et sera une caractéristique de l'ensemble (P.C.C. - objets).

IID2b Distance moyenne et espérance mathématique.

L'espérance mathématique de la distance entre deux objets s'écrit :

$$E(d) = \sum_{d} d \; p(d)$$

où p(d), probabilité de parcourir la distance d, est fonction des caractéristiques de communications des objets connectés.

Exemple : soit un réseau où la moitié des objets connectés communiquent avec ceux situés à une distance 2, les autres ne communiquent qu'avec ceux situés à une distance 4. Si le réseau relie tous les objets par des chemins de longueur 1, il vient :

$$\bar{d} = 1 \qquad E(d) = 3.$$

IID2c Calcul de l'espérance mathématique sur un P.C.C.

Cette section présente succintement les résultats dont la justification est donnée en Annexe III.

a) Réseau uniforme.

Un réseau de n^2 cellules auxquelles sont connectés n^2 objets est dit uniforme si un objet (i, j) communique avec un objet (k, ℓ) avec la probabilité :

$$P_{ij}^{k\ell} = \frac{1}{n^2 - 1} \quad \forall i,j,k,\ell \text{ t.q. } (i, j) \neq (k, \ell)$$

La relation "l'objet (i,j) équivaut à l'objet (k, ℓ) si $N_{ij}^d = N_{k\ell}^d$" (où N_{ij}^d est le nombre d'objets situés à la distance d de l'objet (i, j)) est une relation d'équivalence. Cette relation d'équivalence détermine une partition de l'ensemble des cellules en K(n) classes d'équivalence.

Le calcul de N_{ij}^d (qui est fonction de i, de j, de d, de la dimension n du réseau) est présenté en annexe III.

Pour un objet (i, j), la probabilité d'effectuer un transfert à la distance d s'écrit :

$$p_{ij}(d) = \frac{N_{ij}^d}{n^2 - 1}$$

Sur l'ensemble du réseau il vient :

$$p_n(d) = \frac{\sum_{i,j} N_{ij}^d}{n^2(n^2 - 1)}$$

D'où :

$$E_n(d) = \frac{1}{n^2(n^2 - 1)} \sum_{d \in D_n} \{d \sum_{i,j} N_{ij}^d\}$$

où D_n est le domaine de d. $D_n = \{2, 3, \ldots, 2n-1\}$. Ce qui aboutit, dans le cas d'un réseau uniforme au résultat :

$$\underline{\underline{E_n(d) = \frac{2n}{3} + 1 = \bar{d}_n}}$$

b) Réseau à principe de localité linéairement décroissante.

La loi de probabilité de communication s'écrit :

$P_o(d) = \dfrac{2n - d}{(n-1)(2n-1)}$ = probabilité de communiquer à la distance d de l'objet (i, j).

Cette loi a été distordue dans les conditions de la simulation (cf. Annexes I et III), d'où

$$P_{ij}(d) = \sum_{k=o}^{\infty} P_o(d) \left\{ \sum_{d=M_{ij}+1}^{2n-1} P_o(d) \right\}^k$$

où M_{ij} est la distance maximale accessible à partir d'un objet (i, j).

Sur l'ensemble du réseau, il vient :

$$P(d) = \frac{1}{n^2} \sum_{C_{k'} \in C} [\#C_{k'}] \times \left(\sum_{k=0}^{\infty} P_o(d) \left\{ \sum_{M_{C_{k'}}+1}^{2n-1} P_o(d) \right\}^k \right)$$

où $M_{C_{k'}}$ est la distance maximale accessible à partir d'un élément de la classe $C_{k'}$, de l'ensemble C des classes d'équivalence définies en a).

Ceci aboutit à l'espérance mathématique de la distance :

$$E_n(d) = 1 + 0{,}58n$$

c) Réseau à principe de localité "en escalier".

Principe : une proportion W des communications issues d'un objet s'effectue à une distance inférieure ou égale à L, dite distance de localité. Dans la zone ainsi définie, les communications sont équiprobables quelle que soit la distance. Une proportion 1-W des communications a lieu à une distance comprise entre L+1 et 2n-1 (2n-1 = max $\{D_n\}$). Dans ce domaine les communications sont équiprobables quelle que soit la distance.

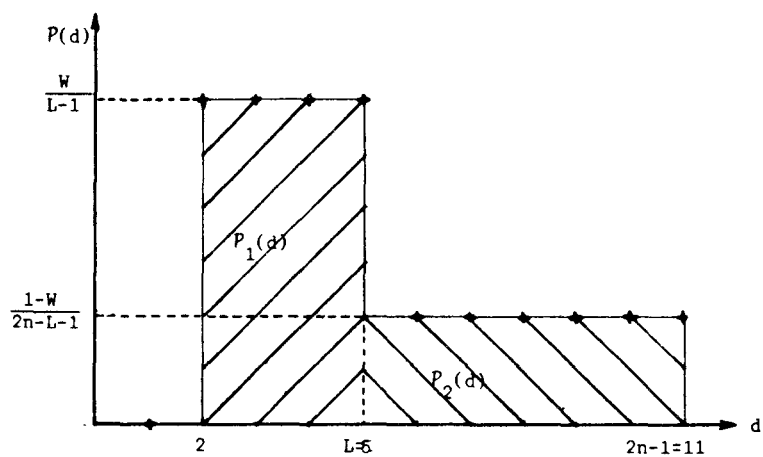
La loi théorique s'écrit donc :

$$P(d) = P_1(d) \cup P_2(d)$$

où

$$P_1(d) = \frac{W}{L-1} \qquad \forall d \in [2, L] \qquad (d \text{ entier})$$

$$P_2(d) = \frac{1-W}{2n-L-1} \qquad \forall d \in]L, 2n-1] \ (d \text{ entier})$$



Exemple dans le cas où L=5
n=6

La loi ayant été distordue dans les conditions de la simulation, la loi simulée a été (cf. Annexes I et III) :

$$P'_{ij}(d) = \sum_{k=0}^{\infty} P(d) \left\{ \sum_{d=M_{ij}+1}^{2n-1} P_2(d) \mid N_{ij}^{d} = 0 \right\}^{k}$$

Sur l'ensemble du réseau il vient :

$$P'(d) = \frac{1}{n^2} \sum_{C_{k'} \in C} [\#C_{k'}] \times \left(\sum_{k=0}^{\infty} P(d) \left\{ \sum_{d=M_{C_{k'}}+1}^{2n-1} P_2(d) \right\}^{k} \right)$$

Ceci a permis de calculer E(d) dont une approximation linéaire s'écrit :

$$E(d) = 0{,}5\ L + 5\ W + 6 \quad (n = 12)$$

L'influence prépondérante de W par rapport à celle de L ainsi mise en évidence traduit le handicap que constituent les communications à grande distance à travers le P.C.C. .

IID3 MODELISATION DU P.C.C.

IID3a Définitions et rappels.

Un objet connecté au P.C.C. se caractérise par le taux λ, probabilité théorique de créer un message durant un cycle élémentaire de l'objet. Les objets sont directement connectés au P.C.C., sans interface ni file d'attente (tout au moins dans une première analyse). Cousin [COU76] a montré que le P.C.C. a, hors d'un phénomène transitoire de démarrage, un fonctionnement

- stationnaire
- ergodique
- sans interblocage en l'absence de panne.

Les valeurs moyennes suivantes, caractéristiques du fonctionnement du P.C.C. seront étudiées :

$\bar{I}_p$: nombre moyen d'injections dans le P.C.C. durant un cycle élémentaire d'un objet

$\bar{C}_R$: nombre moyen de conflits durant un cycle élémentaire du P.C.C.

$\bar{T}_R$: nombre moyen de transferts par cycle du P.C.C.

$\overline{Ex_p}$: nombre moyen d'extractions par cycle d'objet

$\bar{V}$: durée moyenne de transit (en cycles d'objet)

$\bar{C}_M$: nombre moyen de registres M occupés en fin de cycle P.C.C.

$\bar{\Sigma}$: nombre moyen de messages présents dans le P.C.C.

Les variables $\bar{C}_p$ et $\bar{T}_p$, liées à $\bar{C}_R$ et $\bar{T}_R$, peuvent également être définies :

$$\bar{C}_p = R\ \bar{C}_R$$

$$\bar{T}_p = R\ \bar{T}_R$$

où R est le rapport $\frac{\text{cycle de base d'un objet}}{\text{cycle P.C.C.}}$; $R \in \mathbb{N}$.

Ces valeurs moyennes sont liées par des relations fondamentales exprimées par Cousin (R=1) :

$$\bar{\Sigma} = \bar{I}_R + \bar{C}_R + \bar{T}_R + \bar{Ex}_p \qquad \bar{I}_R = \overline{Ex_p}$$

$$\Rightarrow \quad \bar{\Sigma} = 2\bar{I}_R + \bar{C}_R + \bar{T}_R$$

$\bar{T}_R = \bar{I}_R(\bar{d}_\gamma + 1)$ ce qui correspond avec nos notations à

$$\bar{T}_R = \bar{I}_R.E(d) \Longrightarrow \bar{\Sigma}_R = \bar{I}_R\,[E(d) + 2] + \bar{C}_R$$

$\hat{\lambda}_n = \frac{1}{2+E(d)}$ où $\hat{\lambda}_n$ est la valeur critique de λ au delà de laquelle le réseau est saturé.

$$\text{si } \lambda < \hat{\lambda}_n, \quad \bar{I}_R \approx \lambda n^2$$
$$\text{si } \lambda \geq \hat{\lambda}_n, \quad \bar{I}_R \approx \hat{\lambda}_n\, n^2$$

Ces relations peuvent se généraliser au cas où $R \geq 1$ en posant :

$$\bar{\Sigma}_p = \bar{I}_p + \bar{C}_p + \bar{T}_p + \overline{Ex_p} = 2\bar{I}_p + R\,\bar{C}_R + R\,\bar{T}_R$$

IID3b Matrice des probabilités de transfert (R=1)

Il s'agit d'exprimer la probabilité qu'un message présent dans un registre donné soit à transférer vers un autre registre. Cousin a montré qu'un message présent dans le registre M demande un transfert vers le registre R_x avec la probabilité :

$$E_x = \frac{1}{2}\,\frac{1}{n+1} \qquad (x = N \text{ ou } S)$$

$$E_x = \frac{1}{2}\,\frac{1}{n+1} \qquad (x = O \text{ ou } E)$$

Ces relations sont vérifiées en l'absence de principe de localité, il s'agit de les généraliser dans les limites de nos hypothèses.

Une matrice M représente les probabilités étudiées :

$$M = \begin{bmatrix} P_{NN} & 0 & 0 & 0 & P_{NR} \\ P_{EN} & P_{EE} & P_{ES} & 0 & P_{ER} \\ 0 & 0 & P_{SS} & 0 & P_{SR} \\ P_{ON} & 0 & P_{OS} & P_{OO} & P_{OR} \\ P_{MN} & P_{ME} & P_{MS} & P_{MO} & 0 \end{bmatrix}$$

où M_{ij} = probabilité d'aller du registre i au registre j ; les termes nuls correspondent aux transferts impossibles par structure du P.C.C.

◊ Relations remarquables :

par symétrie des transferts, les relations suivantes sont vérifiées :

$$P_{MN} = P_{MS} \qquad P_{ME} = P_{MO}$$

$$P_{NR} = P_{SR} \qquad P_{ER} = P_{OR}$$

$$P_{ES} = P_{OS} = P_{ON} = P_{EN} \qquad P_{NN} = P_{SS} = P_{EE} = P_{OO}$$

$$\frac{P_{MN}}{P_{MO}} = \frac{P_{OR}}{P_{NR}} = \frac{P_{ER}}{P_{NR}} = \frac{P_{ER}}{P_{SR}} = \frac{P_{OR}}{P_{SR}} = \frac{P_{MN}}{P_{ME}} = \frac{P_{MS}}{P_{MO}} = \frac{P_{MS}}{P_{ME}}$$

◊ Calcul des termes de M.

. Ligne de M

Un objet émet vers 2L (L-1) objets dont 2(L-1) sont situés sur la même colonne

$$\Rightarrow P_{MN} + P_{MS} = \frac{2\ (L-1)}{2L\ (L-1)} = 2P_{MN} = 2P_{MS} \Rightarrow \boxed{P_{MN} = P_{MS} = \frac{1}{2L}}$$

Les autres objets sont servis après un cheminement horizontal puis vertical. D'où :

$$P_{MO} = P_{ME} = \frac{L-1}{2L}.$$

. Ligne de N

Sur la i^{e} rangée (en partant de la rangée de l'émetteur), il y a 2(L-i)-1 déplacements N-R, soit au total :

$\sum_{i=1}^{L-1} [2(L-i) - 1] = (L-1)^2$ déplacements N-R.

Sur la i^{e} rangée (i > 1), il y a 2(L-i)-1 déplacements N-N, soit au total

$\sum_{i=2}^{L-1} [2(L-i) - 1] = L^2 - 4L + 4$ déplacements N-N.

De ces calcùls, on déduit :

$$P_{NR} = \frac{(L-1)^2}{2L^2 - 6L + 5} = P_{SR}$$

$$P_{NN} = \frac{L^2 - 4L + 4}{2L^2 - 6L + 5}$$

. Autres termes

Le rapport $\frac{P_{MN}}{P_{MO}} = \frac{P_{OR}}{P_{NR}}$ ainsi que la somme des termes d'une même ligne (égale à 1) impliquent :

$$P_{OR} = P_{ER} = \frac{L-1}{2L^2 - 6L + 5}$$

$$P_{ON} = \frac{L^2 - 3L + 2}{2(2L^2 - 6L + 5)}$$

0 Forme générale de M

$$M = \begin{bmatrix} a & o & o & o & e \\ b & a & b & o & f \\ o & o & a & o & e \\ b & o & b & a & f \\ c & d & c & d & o \end{bmatrix}$$

où

$$a = \frac{L^2 - 4L + 4}{2L^2 - 6L + 5}$$

$$b = \frac{L^2 - 3L + 2}{2(2L^2 - 6L + 5)}$$

$$c = \frac{1}{2L}$$

$$d = \frac{L-1}{2L}$$

$$e = \frac{(L - 1)^2}{2L^2 - 6L + 5}$$

$$f = \frac{L - 1}{2L^2 - 6L + 5}$$

La matrice M', déduite de M en ajoutant une ligne de zéros (registre R) et une colonne de zéros (registre M) sera utilisée par la suite.

0 Exemples de matrices de probabilités de transfert.

L=2

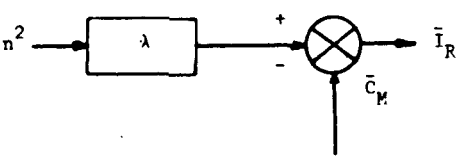
$$M' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0.25 & 0.25 & 0 & 0 \end{pmatrix}$$

L=4

$$M' = \begin{array}{c} \begin{matrix} N & E & S & O & R & M \end{matrix} \\ \begin{pmatrix} 0.308 & 0 & 0. & 0 & 0.692 & 0 \\ 0.230 & 0.308 & 0.230 & 0 & 0.230 & 0 \\ 0 & 0 & 0.308 & 0 & 0.692 & 0 \\ 0.230 & 0 & 0.230 & 0.308 & 0.230 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.125 & 0.375 & 0.125 & 0.375 & 0 & 0 \end{pmatrix} \end{array}$$

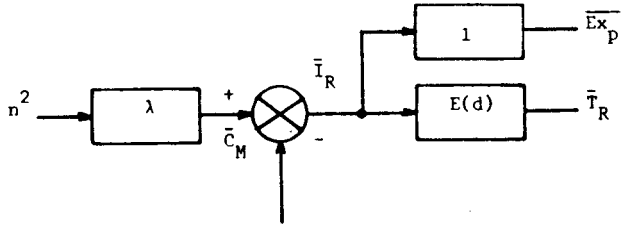
IID3c Modélisation par contre-réaction (R=1)

Les n^2 objets connectés au réseau étant susceptibles d'émettre au taux λ, les injections effectivement réalisées sont, parmi celles qui ont été tentées, celles qui étaient possibles parce que le registre M de la cellule correspondante était libre. Le schéma correspondant s'écrit :

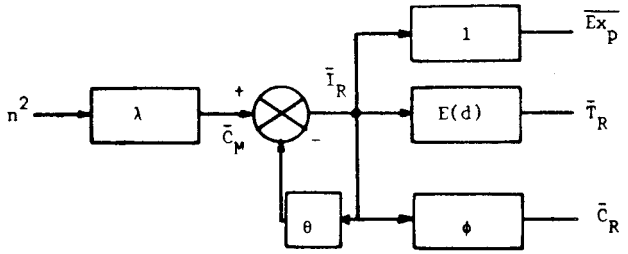


Le nombre de transferts $\bar{T}_R$ est relié au nombre d'injections $\bar{I}_R$ par la relation $\bar{T}_R = \bar{I}_R.E(d)$.

D'autre part, le nombre d'extractions est égal au nombre d'injections ; le schéma se complète donc :



Il reste à indiquer sur le schéma comment sont obtenus le nombre moyen de conflits $\bar{C}_R$ et le nombre moyen de registres émetteurs occupés en début de cycle ($\bar{C}_M$). Un schéma global du type suivant a été recherché :



Sur un tel schéma, les relations s'écrivent :

$$\bar{C}_R = \phi(\bar{I}_R)$$

$$\bar{C}_M = \theta(\bar{I}_R)$$

$$\bar{I}_R = \overline{Ex_p} = \frac{\bar{T}_R}{E(d)} = \lambda n^2 - \theta(\bar{I}_R)$$

où θ et ϕ sont des fonctions de transfert dépendantes de $\bar{I}_R$. Ces fonctions sont à définir ; accessoirement la relation liant $\bar{T}_R$ à $\bar{I}_R$ sera calculée par une nouvelle méthode.

IID3d Calcul des relations existant dans le modèle.

Soit le réseau à l'instant t, un cycle venant de s'achever (les conditions du régime transitoire sont terminées) ; le vecteur $\vec{\alpha}$, vecteur de probabilité d'occupation des différents registres d'une cellule idéale possédant les caractéristiques moyennes des cellules du P.C.C. s'écrit :

$$\vec{\alpha} = (\alpha_N, \alpha_E, \alpha_S, \alpha_O, \alpha_R, \alpha_M)$$

Les injections dans le P.C.C. se font avec la probabilité (par cellule) : $\tau = \frac{\bar{I}_R}{2}$. Soit $\vec{\tau}$ le vecteur (0, 0, 0, 0, 0, τ). Après les injections dans le P.C.C., le vecteur des probabilités d'occupation devient $\vec{\alpha}_i = \vec{\alpha} + \vec{\tau}$.

Après les transferts, le vecteur des probabilités d'occupation devient, étant données la stationnarité et l'ergodicité :

$$\vec{\alpha}_t = \vec{\alpha} + \vec{\tau}'$$

où $\vec{\tau}'$ est le vecteur (0, 0, 0, 0, τ, 0) puisque $\bar{I}_R = \overline{Ex_p}$. Après les extractions, le vecteur des probabilités d'occupation est de nouveau égal à $\vec{\alpha}$.

0 Propriétés remarquables de $\vec{\alpha}$.

De par la symétrie du réseau, les registres N et S d'une part, O et E d'autre part, sont équivalents :

$$\alpha_N = \alpha_S \qquad \alpha_E = \alpha_O$$

Le registre R est systématiquement vidé en fin de cycle, donc

$$\alpha_R = 0$$

0 Calcul de $\vec{\alpha}$.

Les probabilités d'occupation après les transferts et conflits sont liées aux probabilités d'occupation avant transferts et conflits :

$$\vec{\alpha}_t = \vec{\alpha} + \vec{\tau}' = f(\vec{\alpha}_i) = f(\vec{\alpha} + \vec{\tau})$$

Caractérisation de la fonction f :

$\alpha_S = \alpha_N = \alpha_N\ a\ \alpha_N + \alpha_N\ a(1-\alpha_N) + \alpha_E\ b(1-\alpha_N) + \alpha_E\ b(1-\alpha_N)$ $+ \alpha_N\ \tau e + (\alpha_M + \tau)\ c\ (1-\alpha_N)$
$\alpha_O = \alpha_E = \alpha_E\ a\ \alpha_E + \alpha_E\ a(1-\alpha_E) + \alpha_E\ b\ \alpha_N + \alpha_E\ b\ \alpha_N$ $+ \alpha_E\ f\ \tau + (1-\alpha_E)\ d\ (\alpha_M + \tau)$
$\tau = \alpha_N\ e\ (1-\tau) + \alpha_E\ f\ (1-\tau) + \alpha_N\ e\ (1-\tau) + \alpha_E\ f\ (1-\tau)$
$\alpha_M = (\alpha_M + \tau)\ c\ \alpha_N + (\alpha_M + \tau)\ d\ \alpha_E + (\alpha_M + \tau)\ c\ \alpha_N + (\alpha_M + \tau)\ d\ \alpha_E$

Plus simplement :

(1) $\alpha_S = \alpha_N = a\,\alpha_N + 2\,b\,\alpha_E(1-\alpha_N) + \alpha_N\,\tau\,e + (1-\alpha_N)\,c\,(\alpha_M + \tau)$

(2) $\alpha_O = \alpha_E = a\,\alpha_E + 2\,b\,\alpha_N\,\alpha_E + \alpha_E\,\tau\,f + (1-\alpha_E)\,d\,(\alpha_M + \tau)$

(3) $\tau = 2\,(e\,\alpha_N + f\,\alpha_E)\,(1 - \tau)$

(4) $\alpha_M = 2\,(c\alpha_N + d\,\alpha_E)\,(\alpha_M + \tau)$

Ce système de quatre équations non linéaires à trois inconnues (α_N, α_E, α_M) peut être résolu numériquement mais il peut être résolu dans le cas général.

La somme de (1) et (2) donne :

$$\alpha_N + \alpha_E = a(\alpha_N + \alpha_E) + 2\,b\,\alpha_E + \tau(e\alpha_N + f\alpha_E) + (\alpha_M + \tau)\cdot\,[c + d - c\alpha_N - d\alpha_E]$$

Les relations (3), (4) ainsi que $c + d = \frac{1}{2}$ permettent d'obtenir :

$$\alpha_N + \alpha_E = a(\alpha_N + \alpha_E) + 2b\alpha_E + \frac{\tau^2}{2(1-\tau)} + \frac{\tau}{2}$$

$$\Longrightarrow \quad \boxed{\alpha_N(1-a) + \alpha_E(1-a-2b) = \frac{\tau}{2(1-\tau)}}$$

$$\boxed{\alpha_N e + f\alpha_E = \frac{\tau}{2(1-\tau)}}$$

Ces deux équations linéaires sont résolues par calcul des déterminants ; les relations e = 1-a et f = 1-a-2b impliquent que les déterminants

sont tous nuls, d'où :

$$\alpha_N = \alpha_E = \alpha_S = \alpha_0$$

D'où finalement :

$\alpha_N = \alpha_E = \alpha_S = \alpha_0 = \frac{\tau}{2(1-\tau)(e+f)}$
$\alpha_R = 0$
$\alpha_M = \frac{\tau^2}{2(1-\tau)(e+f) - \tau}$

◊ Utilisation de $\vec{\alpha}$ dans le calcul des caractéristiques du P.C.C. .

Pour qu'un conflit se produise dans le P.C.C., il faut

1) que le registre fournissant l'information soit occupé

2) que le transfert demandé concerne ce registre "fournisseur"

3) vers un registre "client" occupé.

De même, le transfert résulte

1) d'un registre "fournisseur" occupé

2) d'un choix de transfert de ce registre vers

3) un registre "client" non occupé.

Le vecteur des probabilités de conflit par registre fournisseur s'écrit :

$$\vec{\gamma} = (\vec{\alpha} + \vec{\tau}) \quad [M' \times (\vec{\alpha} + \vec{\tau}')]$$

où le produit $M' \times (\vec{\alpha} + \vec{\tau}')$ est le produit du vecteur colonne $(\vec{\alpha} + \vec{\tau}')$ par la matrice M' définie précédemment. Le produit par $\vec{\alpha} + \vec{\tau}$ donne pour chaque registre fournisseur la probabilité de conflit.

En remplaçant les composantes de $\vec{\alpha}$, $\vec{\tau}$, $\vec{\tau}'$, M' par leur valeur il vient :

$$\vec{\gamma} = \{a\alpha_N^2 + \alpha_N e\tau,\ (a + 2b)\ \alpha_N^2 + \alpha_N f\tau,\ a\alpha_N^2 + \alpha_N e\tau,\ (a + 2b)\ \alpha_N^2 + \alpha_N f\tau,\ 0,\ \alpha_N(\alpha_M + \tau)\}$$

La probabilité globale de conflit sur les registres d'une cellule s'écrira (en considérant les conflits sur des registres distincts comme des événements indépendants) :

$$\Gamma = \sum \gamma_i - \sum_{i\neq j} \gamma_i\gamma_j + \sum_{i\neq j\neq k} \gamma_i\gamma_j\gamma_k - \sum_{i\neq j\neq k\neq \ell} \gamma_i\gamma_j\gamma_k\gamma_\ell + \sum_{\substack{i\neq j\neq k\neq \ell \\ \neq m}} \gamma_i\gamma_j\gamma_k\gamma_\ell\gamma_m$$

ce qui s'approxime par le premier terme lorsque $\gamma_i\gamma_j \ll \gamma_i$, donc lorsque les conflits sont faibles :

$$\Gamma \simeq \frac{\tau^2(a+b)}{(1-\tau)^2(e+f)^2} + \frac{\tau^2}{1-\tau} + \frac{\tau^2}{2(1-\tau)(e+f)-\tau} = \Gamma' \quad \gamma_i \ll 1 \quad \forall i$$

Ce calcul permet de caractériser la fonction ϕ définie sur le schéma de contre-réaction.

De la même manière est défini le vecteur des probabilités de transfert par registre fournisseur :

$$\vec{\phi} = (\vec{\alpha} + \vec{\tau}) \quad [M' \times [\vec{u} - (\vec{\alpha} + \vec{\tau}')]]$$

où $\vec{u}$ est le vecteur unitaire.

En remplaçant les composantes de $\vec{\alpha}$, $\vec{\alpha}'$, $\vec{\tau}$, $\vec{\tau}'$, $\vec{u}$, M' par leur valeur il vient :

$$\vec{\phi} = \{\alpha_N - a\alpha_N^2 - e\alpha_N\tau,\ \alpha_N - (a+2b)\alpha_N^2 - \alpha_N f\tau,\ \alpha_N - a\alpha_N^2 - e\alpha_N\tau,$$
$$\alpha_N - (a+2b)\alpha_N^2 - \alpha_N f\tau,\ 0,\ (\alpha_M + \tau)\ (1-\alpha_N)\}$$

Ce qui vérifie la relation $\vec{\phi} + \vec{\gamma} = \vec{\alpha} + \vec{\tau}$ car s'il y a eu conflit ou transfert, c'est parce que le registre était occupé ! La probabilité globale de transferts sur les registres d'une cellule s'écrit :

$$\psi = \sum_i \phi_i - \sum_{i\neq j} \phi_i\phi_j + \sum_{i\neq j\neq k} \phi_i\phi_j\phi_k - \sum_{i\neq j\neq k\neq \ell} \phi_i\phi_j\phi_k\phi_\ell + \sum_{\substack{i\neq j\neq k\neq \ell \\ \neq m}} \phi_i\phi_j\phi_k\phi_\ell\phi_m$$

avec bien évidemment :

$$\psi = 4\alpha_N + \alpha_M + \tau - \Gamma$$

Cette probabilité globale peut s'approximer par le premier terme si $\Phi_i \ll 1 \ \forall i$, d'où

$$\psi = \frac{2\tau}{(1-\tau)(e+f)} + \frac{2\tau(1-\tau)\ (e+f)}{2(1-\tau)\ (e+f) - \tau} - \Gamma' = \psi'$$

Une nouvelle formulation de la relation liant $\bar{T}_R$ à $\bar{I}_R$ est ainsi écrite ; les applications numériques montrent que la nouvelle relation équivaut pratiquement à $\bar{T}_R = E(d)\ \bar{I}_R$.

La dernière variable utilisée sur le schéma de contre-réaction est $\bar{C}_M$; elle représente les registres émetteurs qui n'ont pu être vidés au cours d'un cycle à cause de conflits. γ_M représente cette éventualité, d'où :

$$\bar{C}_M = n^2\ \alpha_N(\alpha_M + \tau)$$

$$\Longrightarrow \quad \bar{C}_M = \frac{n^2\tau^2}{2(1-\tau)(e+f) - \tau}$$

Cette relation caractérise la fonction θ définie sur le schéma de contre-réaction.

IID3e Application - Vérification - Validité.

Grâce aux simulations effectuées, une valeur de τ a pu être mesurée dans différents cas. Ces valeurs ont été utilisées pour vérifier la validité du modèle pour différentes valeurs de L et λ sur un réseau 12 × 12.

	L=2 W=0.95 λ = 0.25	L=4 W=0.95 λ = 0.10	L=4 W=0.95 λ = 0.25	L=4 W=0.95 λ = 0.50
τ mesuré	0.196	0.088	0.188	0.199
τ recalculé	0.237	0.095	0.223	0.469
e	1	0.692	0.692	0.692
f	1	0.230	0.230	0.230
α_N	0.0609	0.0523	0.1255	0.1347
α_M	0.0127	0.0049	0.0270	0.0310
γ_M	0.0127	0.0049	0.0270	0.0310
$\bar{C}_M$ calculé	1.83	0.699	3.88	4.46
$\bar{C}_M$ simulé	7.804	1.752	8.232	43.284
Γ	Γ' = 0.060	Γ' = 0.011	Γ' = 0.104	Γ' = 0.120
$\bar{C}_R$ calculé	8.64	1.584	14.98	17.28
$\bar{C}_R$ simulé	10.776	3.748	14.53	93.812
ψ	ψ = 0.392	ψ' = 0.291	ψ' = 0.613	ψ' = 0.649
$\bar{T}_R$ calculé	56.49	41.92	88.27	93.43
$\bar{I}_R$. E(d)	73.25	49.45	84.91	112.28
$\bar{T}_R$ simulé	66.368	46.23	79.01	105.508

Conclusions.

a) La seconde ligne indique la valeur de τ recalculée, c'est-à-dire obtenue par

$$\tau = \frac{\lambda n^2 - \bar{C}_M}{n^2} = \lambda - \gamma_M$$

b) Le modèle n'est valable que sous certaines hypothèses, en particulier :

- réseau infini, or l'application à été faite pour un réseau 12 × 12 où les effets de bords ne sont pas négligeables

- localité parfaite, c'est-à-dire W = 1.00. L'application a été faite pour W = 0.95 d'où une erreur d'au moins 5 % (l'importance de W a été soulignée dans l'étude des métriques).

c) Les simulations n'ont été faites que sur un nombre limité de cycles du P.C.C. (1000 au maximum, dont plus d'une centaine pour la phase transitoire), leurs résultats sont donc entachés d'erreur.

d) Dans tous les cas, Γ' et ψ' ont été prises comme base des calculs or cette approximation n'est justifiée que dans les cas où le P.C.C. est peu chargé (ce qui est faux dans la dernière colonne).

e) Les résultats calculés sont les moins bons pour $\bar{C}_M$, ce qui peut être dû soit à l'influence plus grande des effets de bords sur cette variable, soit à l'influence du programme de simulation (le générateur de variables aléatoires introduit des corrélations "parasites" sur les émissions des cellules balayées successivement).

Malgré ces réserves, les ordres de grandeur obtenus sont généralement bons, surtout si le réseau présente une localité forte (L faible) et une charge peu élevée (non saturation du P.C.C.). La validité des calculs est d'ailleurs corroborée par la nouvelle méthode de calcul de $\bar{T}_R$ dont les résultats sont voisins de $E(d).\bar{I}_R$.

IID4 CONSEQUENCES ET GENERALISATION DU MODELE.

Le modèle précédent était établi dans le cas le plus simple, il s'agit donc de le modifier afin de le rendre plus proche d'un réseau réel, c'est-à-dire essentiellement :

- de le généraliser au cas où $R \geq 1$
- de réaliser l'exclusion mutuelle en l'absence d'interface (un objet reçoit ou émet à un instant donné mais il ne fait que rarement les deux)
- l'insertion de files en interface entre le P.C.C. et les objets connectés.

IID4a Généralisation au cas où $R \geq 1$.

Dans le cas où R (entier) est supérieur à 1, les injections et extractions de et vers le P.C.C. ne sont possibles que durant un cycle P.C.C., simultanément avec des transferts ; durant les autres cycles simultanés à un cycle objet seuls des transferts sont exécutés.

Le calcul du modèle nécessite donc le calcul des composantes de $\vec{\alpha}$ après chaque cycle P.C.C. La séquence $(\vec{\alpha}_o, \vec{\alpha}_o + \vec{\tau}, \vec{\alpha}_o + \vec{\tau}', \vec{\alpha}_o)$ décrivant le cas où R = 1 se généralise donc en une séquence plus complexe

$$\vec{\alpha}_o,\ \vec{\alpha}_o + \vec{\tau},\quad \vec{\alpha}_1,\quad \vec{\alpha}_2,\ \ldots,\ \vec{\alpha}_R = \vec{\alpha}_o + \vec{\tau}',\quad \vec{\alpha}_o$$

après injections / après transferts : 1er cycle P.C.C. durant un cycle objet

après transferts : 2e cycle P.C.C.

après transferts / après extractions : R^e cycle P.C.C.

1 cycle objet

R valeurs de $\vec{\alpha}$ seront donc calculées, et donc 2R inconnues ; ces vecteurs sont liés par le même type de relations que pour R=1 :

$$\vec{\alpha}_1 = f(\vec{\alpha}_o + \vec{\tau})$$

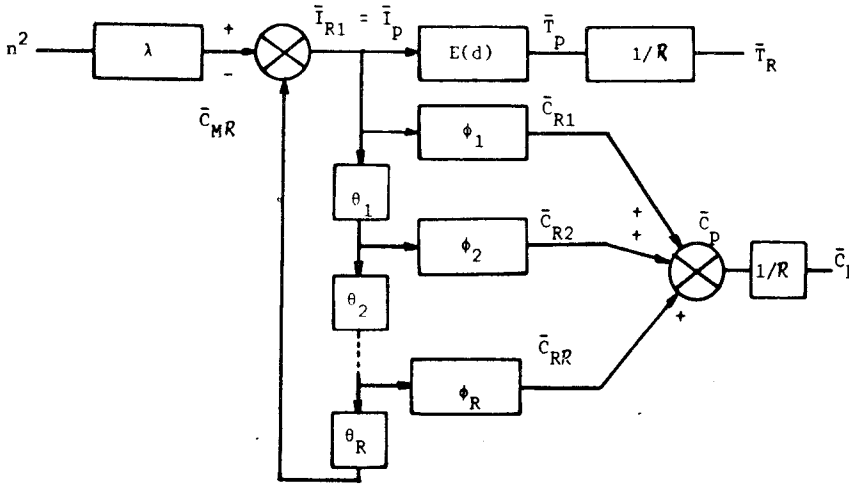
$$\vec{\alpha}_2 = f(\vec{\alpha}_1)$$

etc

$$\vec{\alpha}_R = f^R(\vec{\alpha}_o + \vec{\tau})$$

La connaissance de ces R vecteurs permet de calculer les fonctions ϕ et θ applicables à chaque cycle du P.C.C.

Le schéma de contre-réaction correspondant s'écrit :



$\bar{C}_{Mx}$: registres émetteurs occupés après le xième cycle du P.C.C.

$\bar{T}_P = \bar{I}_P \quad E(d) = \bar{I}_{R1} \times E(d)$ puisque les injections ne se font que sur le premier cycle P.C.C.

Les équations générales établies à partir de ce schéma sont :

$$\bar{C}_R = \frac{\bar{C}_P}{R} = \frac{1}{R}\left[\sum_{i=1}^{R} \bar{C}_{Ri}\right] = \frac{1}{R}\left[\sum_{i=1}^{R} \phi_i(\theta^{i-1}(\bar{I}_P))\right]$$

où $\theta^j = \theta_j\ (\theta_{j-1}\ (\ldots(\bar{I}_P)\ldots))$

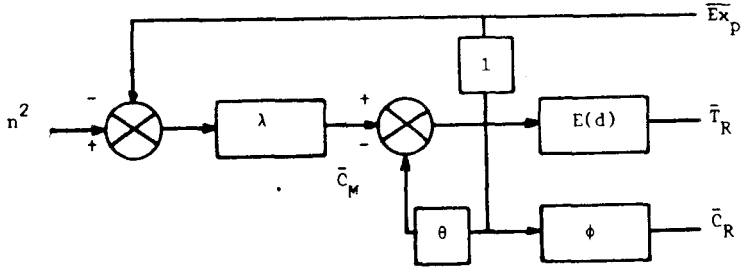
et $\theta^o = 1$

$$\bar{C}_{Mx} = \theta^x(\bar{I}_P)$$

$$\bar{I}_P = \frac{\bar{T}_P}{E(d)} = \frac{R\bar{T}_R}{E(d)} = \lambda n^2 - \theta^R(\bar{I}_P) = R\bar{I}_R.$$

IID4b Exclusion mutuelle.

En l'absence d'interface, il est aisé de prendre en compte l'exclusion mutuelle des injections et réceptions par un objet ; ceci se traduit par le schéma de contre-réaction (R=1) suivant.



Ce schéma suppose que les réceptions sont prioritaires par rapport aux émissions, ce qui semble justifié puisqu'il faut éviter d'encombrer le P.C.C. (risque de blocage).

Les équations caractérisant ce schéma sont :

$$\bar{C}_R = \phi(\bar{I}_R)$$

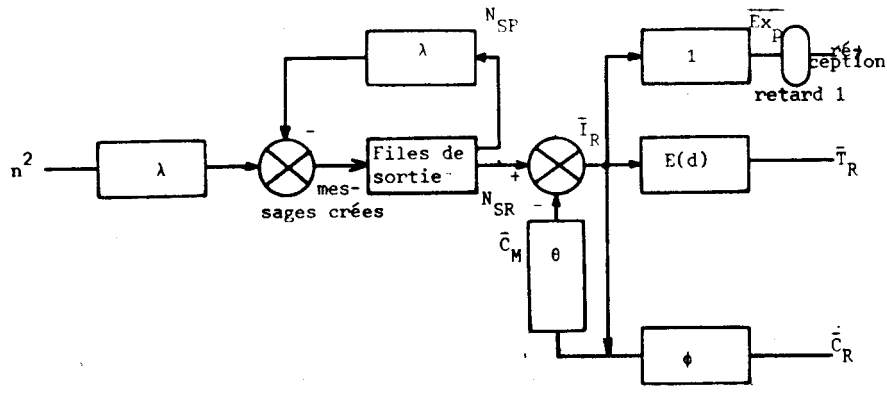
$$\bar{C}_M = \theta(\bar{I}_R)$$

$$\bar{I}_R = \frac{\lambda n^2 - \theta(\bar{I}_R)}{1 + \lambda}$$

Ce schéma est aisément généralisé au cas où R (entier) > 1.

IID4c Insertion d'un interface.

Les simulations ont montré que la présence de files en entrée et en sortie modifiait sensiblement le comportement du système. Dans ce cas, la modélisation est beaucoup plus délicate car elle fait intervenir des interactions entre files d'attente et le P.C.C.(qui lui-même peut être considéré comme un réseau complexe de files d'attente). Le schéma de contre-réaction est difficile à définir rigoureusement, une représentation approximative dans le cas où R=1, sans exclusion mutuelle peut être :



N_{SP} = nombre de files de sortie pleines (le produit par λ suppose que les événements $\left\{\begin{array}{l}\text{- tentative de création}\\ \text{- file de sortie pleine}\end{array}\right\}$ sont indépendants).

Les files de sortie se comportent comme des lignes à retard variable puisque le délai vaut 1 lorsqu'elles sont vides et n'a pas de valeur déterminée lorsqu'elles sont occupées.

N_{SR} = nombre de files de sortie non vides.

Si les réceptions ne sont pas mutuellement exclusives avec les émissions, la file d'entrée se comporte comme une ligne à retard de délai 1.

Remarque

Lorsque le P.C.C. est peu chargé (λ < 0,20 pour R=1), la file de sortie est peu chargée et se comporte comme une ligne à retard de délai unité ; en modifiant la valeur du temps total de parcours (+2), le schéma primitif reste valable.

IIE NOUVELLE DÉFINITION DU PROCESSEUR DE COMMUNICATIONS

IIE1 JUSTIFICATION D'UNE NOUVELLE DEFINITION.

Le réseau conçu par V. Cordonnier [COR75] et R. Cousin [COU75] présente des inconvénients technologiques (nombre de connexions par exemple), économiques (le rapport $\frac{\text{information utile}}{\text{information transférée}}$ est rigide et faible, d'où un rapport (performances/investissement) réduit), et pratiques. Dans cette dernière catégorie citons :

- les problèmes que pose aux objets connectés la réception de plusieurs flots de messages mélangés les uns aux autres

- les liaisons multi-messages. Le transfert par messages de format réduit et fixe ne peut aisément permettre l'échange d'informations en volume variable.

La plupart des caractéristiques du P.C.C. restent toutefois très intéressantes (aspect "pipe-line", modularité, haut niveau de parallélisme, ...) et sont à conserver. La redéfinition du processeur de communications a donc consisté à le modifier afin de pallier aux inconvénients pré-cités et de prendre en compte plus précisément des contraintes annexes telles que la présence d'un interface entre le réseau et les objets connectés, un rapport des vitesses nettement supérieur à l'unité (permis par la technologie), etc...

Le réseau ainsi modifié a été appelé PROCOPE (PROcesseur Cellulaire de COmmunications Par Eléments) et il utilise la notion de trains de messages.

BIS LILLE

IIE2 TRAINS DE MESSAGES : DEFINITION ET JUSTIFICATION.

Un train de messages est une suite de messages élémentaires circulant dans le P.C.C. d'un objet connecté vers un autre ; les messages d'un même train constituent l'information émise à un instant donné par un objet, le train est donc une entité logique du point de vue des objets connectés. Un train T comprendra :

- un message de tête ou "locomotive" (L) utilisé pour le routage dans le réseau

- un nombre quelconque de messages ou "wagons" suivant la locomotive.

IIE2a Routage d'un train.

Lorsqu'un train est envoyé dans le réseau, le premier élément transmis est la locomotive ; lorsque celle-ci est transférée d'un registre à un autre, un aiguillage est positionné. Au cycle suivant, le destinataire

sera fournisseur et positionnera un autre aiguillage. Cet aiguillage "lie" le registre récepteur au registre émetteur ce qui privilégie leur relation d'où :

- l'inutilité d'analyser les wagons lorsque ceux-ci se présentent dans le registre ; leur destinataire est le registre récepteur lié à celui-ci.

- l'impossibilité pour le registre destinataire d'accepter des messages (locomotives ou wagons) de provenance différente du registre fournisseur lié.

Cet aiguillage reste positionné lors du passage des différents wagons et n'est annihilé qu'après transfert du wagon de queue. Une autre locomotive peut alors être prise en compte par le registre auparavant destinataire.

La transmission d'un train se fait donc par transferts successifs de la locomotive qui, en positionnant les aiguillages, ouvre la voie aux wagons qui la suivent. Le chemin suivi par la locomotive (horizontal puis vertical) reste celui défini par R. Cousin et V. Cordonnier car les conditions d'interblocage restent les mêmes (cf. IIF5).

IIE2b Composition d'un train.

Le principe de train permet de réduire le format d'un message élémentaire ; par commodité celui-ci peut ne contenir qu'un octet.

Un message élémentaire correspond soit à une locomotive, soit à un wagon. La locomotive, utilisée pour la définition du chemin dans le réseau, doit contenir

- l'indice de ligne de l'objet récepteur
- l'indice de colonne de l'objet récepteur.

Sur un octet, ceci permet un réseau 16 × 16 et donc 256 objets connectés (au minimum, c'est-à-dire en l'absence de multiplexage).

Un wagon contient soit des informations annexes (adresse de l'objet émetteur, numéro d'ordre du train, octet de contrôle, ...) soit l'information à transmettre.

Afin de permettre une bonne gestion des trains, une dernière information est nécessaire : la spécification du wagon de queue (la locomotive ne doit pas posséder de marque distinctive car c'est le premier message passant par un arc dont l'aiguillage était auparavant non-positionné). Un wagon non terminal ne nécessite aucune marque distinctive.

Cette spécification peut se faire

◊ soit par un codage particulier du message transmis ; cette solution a deux principaux inconvénients : une cellule doit systématiquement décoder un message pour déterminer si c'est un wagon de queue, d'où une lenteur et une complexité de la cellule accrues ; une combinaison de l'octet est inutilisable pour l'échange d'informations. Si un train transmet k octets utiles aux objets, il faut une locomotive et un wagon de queue, d'où $\frac{\text{Information utilisée par le réseau}}{\text{Information transmise}} = \frac{2}{k+2}$

◊ soit par adjonction d'une information supplémentaire, associée à chaque message ; ce bit de suite est par exemple à 1 si le message n'est par terminal, à 0 si le message est terminal. Cette solution en rend le traitement aisé par une cellule, ne nous prive d'aucune combinaison sur un octet, mais impose de concevoir le réseau avec des registres de 9 bits. Si un train transmet k octets utiles, le rapport $\frac{\text{Information utilisée par le réseau}}{\text{Information transmise}} = \frac{k+9}{9k+9}$. Dès lors que k excède 7, cette solution devient plus coûteuse en quantité relative d'information utilisée par le réseau. Elle a cependant été choisie car plus souple et plus performante.

IIE2c Compatibilité avec le P.C.C. défini par R. Cousin.

Les principes directeurs décrits au début de ce chapitre (modularité, extensibilité, flexibilité, uniformité, etc ...) restent vérifiés par le réseau PROCOPE. Les propriétés du graphe de connexions restent vérifiées puisque le réseau PROCOPE peut utiliser le même graphe des liaisons que le P.C.C. .

Kleinrock dans [KLE76] définit la commutation de paquets comme un cas particulier de commutation de messages où un message est découpé en paquets transmis consécutivement par le réseau de communication. Ceci permet d'utiliser l'effet "pipe-line", de réduire le délai global de transmission, de réduire les moyens de stockage présents à un noeud ... Le réseau PROCOPE se rapproche de ce principe de commutation, mais également de la commutation de circuits (ou de lignes) puisque les aiguillages restent positionnés durant le passage d'un train.

Dans le cas de trains de longueur importante transmis entre objets peu éloignés, il y a même quasiment commutation de circuits (ou de lignes). La commutation adoptée devient donc un compromis entre les deux types de commutation permettant d'en réunir les avantages. Nous l'appelerons *commutation de paquets à routage mémorisé*.

IIE2d Avantages de la transmission par trains.

Les principaux avantages résident dans l'atténuation ou la suppression des inconvénients du P.C.C. décrits précédemment :

- le format d'un message élémentaire est réduit (un octet environ) ceci permet une réduction notable du nombre de connexions et donc une faisabilité et un coût de câblage meilleurs

- le rapport $\frac{\text{information utile aux objets}}{\text{information transmise par le réseau}}$ est nettement amélioré puisque dans le cas du P.C.C. il était de l'ordre de 1/2 ; dans le réseau PROCOPE selon les mêmes hypothèses, il devient $\frac{8k}{9k+18}$ pour une information utile de k octets.

Dès lors que plus de 2 octets sont à transmettre, ce rapport est supérieur à 1/2 et donc le rapport investissement/performance sera amélioré

- la réduction de la taille d'un message devrait permettre la construction d'une cellule moins coûteuse et plus rapide

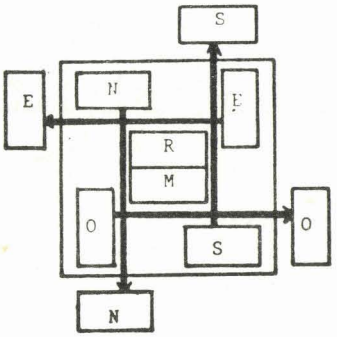
- les trains étant de longueur quelconque, le problème des liaisons multi-messages est résolu et avec lui son corollaire, le problème du tri avant traitement par l'objet destinataire.

IIE2e Inconvénients de la transmission par trains.

Bien que séduisante, la transmission par trains présente certains risques qui sont essentiellement :

- une baisse de performance par blocages temporaires. La S.N.C.F. affiche près de ses passages à niveaux "Attention, un train peut en cacher un autre" ; la maxime à appliquer au réseau PROCOPE serait : "Attention, un train peut en bloquer un autre".

Ce blocage peut se faire soit par la logique de contrôle d'une cellule si celle-ci ne peut traiter plusieurs trains simultanément, soit par la réservation de registres. Le premier risque est résolu en permettant à une cellule, une fois un aiguillage positionné, d'analyser les événements sur d'autres arcs. En effet, sur le graphe des liaisons ci-dessous, on remarque que



les liaisons ne sont pas toutes mutuellement exclusives, puisque n'utilisant ni les mêmes arcs, ni les mêmes noeuds (par exemple les liaisons en trait fort). Au maximum, 5 transferts indépendants sont possibles sur une cellule ; sur l'ensemble du réseau une moyenne de 4 transferts indépendants par cellule est possible. Cette possibilité de transferts peut être mise en oeuvre soit durant un même cycle du réseau (échangeur autoroutier), soit durant des cycles consécutifs (carrefour séquentiel).

Plus dangereux que le blocage par la logique de contrôle est le blocage par réservation des registres puisque lorsqu'un registre a un de ses arcs en cours d'utilisation, il ne peut accepter de transfert venant d'une autre direction ; un ou plusieurs trains peuvent donc être mis en attente forcée durant le temps de passage du premier, quelles que soient leurs longueurs respectives.

- une baisse de performance par inoccupation.

L'un des inconvénients de la commutation de circuits et la sous-utilisation de certaines ressources ; la commutation utilisée étant un compromis entre les deux types de commutation, il est naturel que cet inconvénient apparaisse. Il s'agit dans le cas du réseau PROCOPE de la présence de "trous", c'est-à-dire de registres vides bien que les aiguillages correspondants soient positionnés. Un registre est inoccupé un cycle sur deux, comme sur le P.C.C., mais ce qui est possible (et à éviter) est la présence de plusieurs trous consécutifs le long du chemin emprunté par un train, comme sur le schéma suivant.

(les registres hachurés sont occupés)

Sur cet exemple, 3 registres présentent un "trou" indésirable puisque la ressource est allouée mais anormalement libre.

Ce risque est à prévenir par une adaptation correcte des paramètres tels que la longueur d'un train, le rapport des vitesses réseau-objets connectés (ou interfaces), etc...

- la nécessité de contrôles élaborés.

Sur le P.C.C., l'unité de base est le message et l'erreur est à gérer à ce niveau, d'où des contrôles simples pouvant être câblés, des procédures d'erreur rapides. Sur le réseau PROCOPE, les contrôles se font au niveau de train, d'où leur plus grande complexité et un temps de réponse aux erreurs non négligeable

- la nécessité de files tampon entre les objets et le réseau ; les risques de blocage et d'attente des trains en rendent la présence plus indispensable que sur le P.C.C.

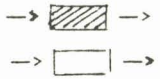
IIE3 DESCRIPTION ET FONCTIONNEMENT D'UNE CELLULE.

Le graphe d'une cellule est celui d'une cellule du P.C.C. ; comme indiqué précédemment beaucoup de propriétés et de caractéristiques de la cellule sont donc identiques à celles étudiées sur le P.C.C. . La principale différence réside dans la logique de contrôle (et le chronogramme associé) décrivant le fonctionnement du réseau PROCOPE.

A l'inverse de la cellule du P.C.C., la solution retenue n'a pas été la structure d'échangeur autoroutier (autorisant des transferts simultanés par une cellule) mais celle du carrefour séquentiel n'autorisant qu'un seul transfert initialisé par une cellule par cycle. Ceci est dû essentiellement à des considérations de coût et de faisabilité ; une cellule plus simple permet également un cycle élémentaire plus court et atténue l'inconvénient de la séquentialité.

Un registre est dit "lié" si un aiguillage le concernant est positionné. Il peut être neutre, lié comme client, comme fournisseur, comme fournisseur et client :

client (registre contenant une locomotive)



fournisseur et client (registre contenant un wagon non terminal ou attendant un wagon)

fournisseur (registre contenant un wagon de queue)

neutre : registre vide et n'attendant aucun wagon.

Le principal choix est de déterminer comment l'enchaînement des registres étudiés se fera afin de garantir l'exclusion mutuelle sur les ressources (registres) tout en conservant de bonnes performances. Il faut donc répondre aux questions suivantes :

- l'enchaînement est-il basé sur les registres fournisseurs ou clients ?
- quel est le critère d'enchaînement ?

Si l'enchaînement se fait sur les registres clients et neutres, il est basé sur les registres vides. Une fois le registre vide choisi, il faut étudier l'état (occupation, aiguillages) de ses fournisseurs possibles, puis choisir parmi ceux-ci (exclusion mutuelle) soit le registre occupé qui est lié au registre de départ, soit un registre fournisseur qui l'a pour destinataire (ce qui nécessite une analyse des contenus des registres fournisseurs) afin de déterminer où doit aller la locomotive). Ensuite, le transfert peut être effectué ; s'il s'agit d'une locomotive un aiguillage est à positionner ; s'il s'agit d'un wagon de queue, un aiguillage est à annihiler. Cette solution présente l'inconvénient de nécessiter de nombreuses transmissions d'information entre cellules avant même d'avoir choisi le registre dont le contenu est à transférer.

Si l'enchaînement se fait sur les registres fournisseurs, il est basé sur les registres occupés. Une fois le registre occupé sélectionné, deux cas se présentent :

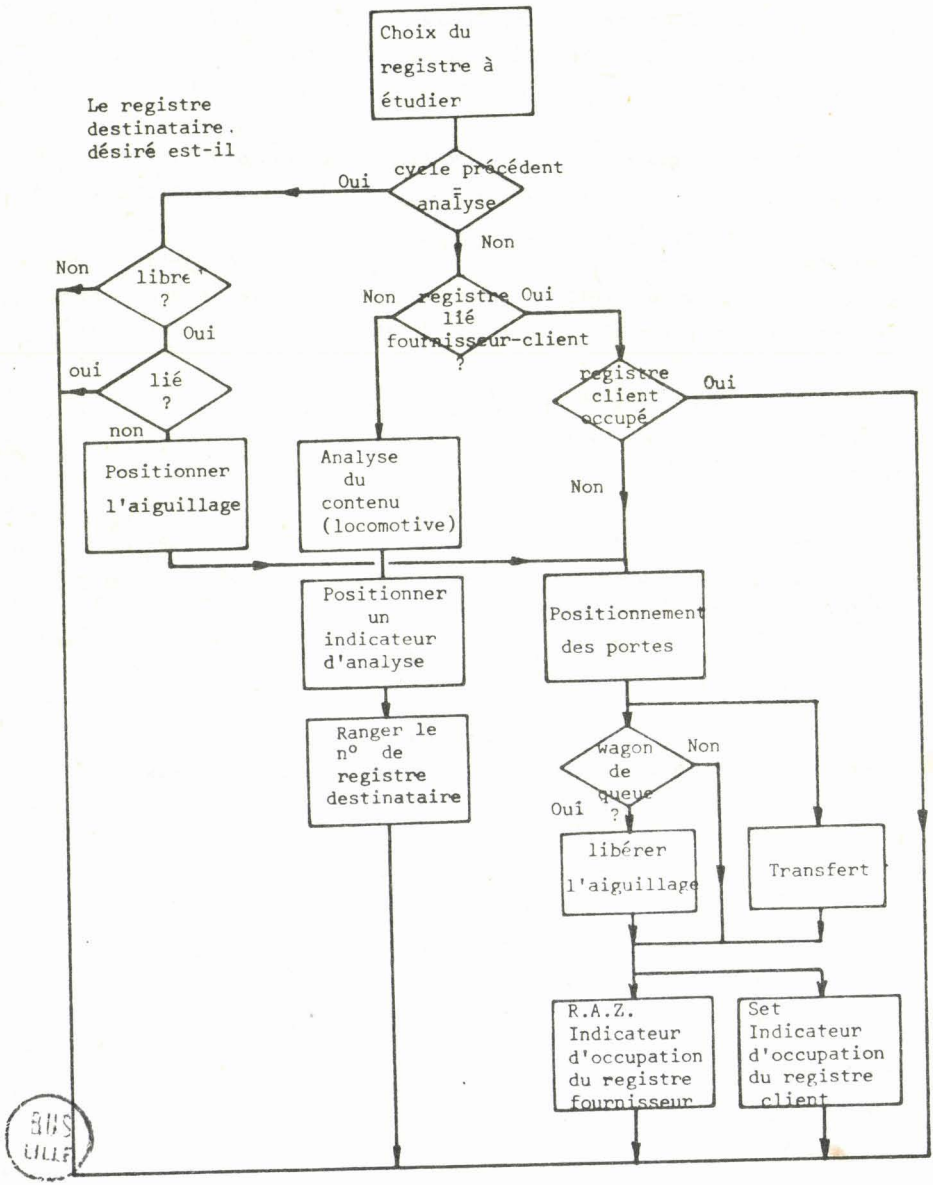
- ce registre est client (contient donc une locomotive). Le registre destinataire est alors calculé ; s'il est neutre, le transfert est possible et l'aiguillage est à positionner. L'exclusion mutuelle est automatique puisqu'un registre d'une cellule ne peut être sollicité que par des registres d'une même cellule. Par exemple, le registre Sud d'une cellule (i,j) ne peut être sollicité par des registres de la cellule (i+1, j)

- ce registre est fournisseur et client, il est donc lié vers un registre destinataire et le transfert est possible si ce dernier est vide.

Cette seconde sòlution nécessite moins de transferts annexes et a été choisie. L'organigramme suivant en précise le fonctionnement, moyennant les remarques suivantes :

- l'annihilation des aiguillages est effectuée lorsque le wagon de queue a été transféré

- une locomotive nécessite deux cycles élémentaires, afin de séparer l'analyse du contenu et les transferts, d'où un raccourcissement du cycle élémentaire d'autant plus profitable que le train est long.


Choix du registre à étudier
Le registre destinataire désiré est-il
Oui
cycle précédent analysé
Non
Non
libre ?
Oui
oui
lié ?
non
Positionner l'aiguillage
Non
registre lié fournisseur-client ?
Oui
Analyse du contenu (locomotive)
registre client occupé
Oui
Non
Positionner un indicateur d'analyse
Ranger le n° de registre destinataire
Positionnement des portes
wagon de queue ?
Non
Oui
libérer l'aiguillage
Transfert
R.A.Z. Indicateur d'occupation du registre fournisseur
Set Indicateur d'occupation du registre client

L'enchaînement des registres occupés peut se faire de différentes méthodes parmi lesquelles :

- le parcours prioritaire.

Dans ce cas, certains registres sont systématiquement étudiés en priorité s'ils sont occupés. L'ordre de priorité peut être fixé par une règle simple telle que favoriser l'extraction des messages proches de leur destination. Ceci amènerait à favoriser les registres Nord et Sud puisqu'en général les messages qu'ils contiennent sont proches de leur terme (par suite de la stratégie des transferts horizontaux puis verticaux). L'ordre décroissant est alors {N, S}, {E, O}, M. Une telle méthode n'est pas acceptable car elle conduit à laisser en attente indéfinie certains registres alors même que les transferts correspondants sont possibles

- le parcours cyclique.

Dans ce cas, les registres sont balayés systématiquement dans un ordre prédéterminé et répété. Afin de limiter les tentatives inutiles, ce parcours est optimisé en n'étudiant que les registres occupés.

La solution adoptée a été le parcours cyclique optimisé où le registre étudié est, dans l'ordre de parcours, le premier registre occupé en partant de la position précédemment étudiée. Ainsi, un registre occupé ne peut attendre plus de 8 cycles avant d'être étudié.

Etat d'une cellule.

L'état d'une cellule est caractérisé d'une part par le contenu des différents registres et des bits de suite qui leur sont associés, d'autre part par des zones d'état associées aux différents registres et définissant :

- l'état d'occupation : vide ou occupé
- l'état de liaison
  * lié ou non lié (peut être confondu avec la spécification suivante si 0 = non lié)

* spécification du registre vers lequel le registre considéré est lié.

Ainsi seul l'état de fournisseur et client est explicitement stocké et les autres états de liaison se définissent par les conditions suffisantes comme suit :

neutre : vide et non-lié
client, fournisseur : occupé

Enfin, l'état d'une cellule est déterminé par la position du dernier registre étudié ainsi que par un indicateur précisant si le cycle précédent était un cycle d'analyse de locomotive (dans ce cas, il y aura tentative de transfert sur ce même registre).

Simulation et modélisation.

Des simulations ont été effectuées afin de vérifier le fonctionnement correct du réseau PROCOPE et d'en évaluer les performances. Ces simulations (cf. Annexe I - Simulations PROCOPE) ont montré la sensibilité du réseau aux divers paramètres (taux d'émission, longueur des trains, etc...). Bien qu'incomplètes elles ont permis d'opter en connaissance de cause pour la réalisation du réseau PROCOPE.

Un modèle probabiliste inspiré du modèle du P.C.C. a également été construit ; il s'est révélé infiniment plus complexe et peu exploitable.

IIF DÉGRADATION DE L'ÉTAT DE PROCOPE.

Cette partie contient la description de diverses malfonctions pouvant affecter le fonctionnement du réseau. Les remèdes à apporter relèvent des protocoles de communication et seront définis avec ceux-ci (Ch. III).

IIF1 PERTE D'INFORMATION DANS LE P.C.C. .

Lors d'un transfert de registre à registre, un message peut être détruit (et il n'en reste aucune trace dans le réseau) soit durant le transfert (la définition de la cellule ne comprend aucune procédure d'acquitement), soit dans le registre récepteur subissant une panne avec effacement de mémoire.

Ces types de pannes peuvent être soit fugitifs, soit permanents. En cas de panne fugitive, l'impact diffère selon le type de message détruit :

- c'est un locomotive ; le wagon suivant sera pris par la cellule comme locomotive ce qui dans un grand nombre de cas entraînera un routage impossible. Cette impossibilité devra être détectée par la cellule.

- c'est un wagon ; le train comprendra un wagon de moins que prévu mais il circulera normalement à travers le réseau. L'erreur ne sera détectable qu'au niveau des interfaces ou des objets connectés.

- c'est un wagon de queue : il y aura concaténation de trains (cf. IIF4).

En cas de panne permanente, des trains peuvent être perdus "corps et biens" dans le réseau ; en conséquence des objets connectés attendront indéfiniment des informations, des commandes ne seront pas acheminées, etc ... Le réseau fonctionne sans difficulté ... mais mal et c'est des objets connectés que doivent provenir la détection et le remède à cette situation.

IIF2 BLOCAGE DU RESEAU PROCOPE.

Cousin avait simulé le blocage d'une cellule du P.C.C. (i.e. la cellule est toujours considéréecomme pleinement occupée). Dans ce cas, le blocage généralisé du P.C.C. est inévitable à brève échéance (quelques 150 cycles environ) ce qui rend l'observation du comportement du réseau et la détection de tout embryon de blocage particulièrement cruciales. Le réseau PROCOPE est plus sensible que le P.C.C. à ce type de panne étant donné que le blocage temporaire d'un train par un autre y est chose courante.

Le blocage peut également prendre naissance parce qu'un registre d'une cellule est toujours considéré comme occupé. Voyons ce problème sur le réseau PROCOPE.

0 Le registre Nord d'une cellule (i,j) est toujours considéré comme occupé. Deux phénomènes se produiront :

a) aucun message ne pourra être transféré vers ce registre Nord. En conséquence, la partie supérieure du réseau (cellules k, ℓ t.q. k < i) se bloquera rapidement (les registres M ne pourront se vider)

b) un train de longueur infinie aura ce registre pour origine. Ce train parasite occupera des registres du réseau. Si ce train est extrait en (i', j'), les registres se trouvant sur le parcours [(i,j), (i',j')] seront toujours occupés ce qui contribuera à paralyser le réseau dans une proportion variable selon (i', j').

0 Le registre Sud d'une cellule est toujours considéré comme occupé : par symétrie du réseau, le résultat est du même type que précédemment.

0 Le registre Est d'une cellule est toujours considéré comme occupé : les deux phénomènes déjà cités se produiront également

1) aucun message ne pourra être transféré vers ce registre, d'où blocage de la ligne du réseau dans sa partie droite (cellules(kℓ) telles que k = i, ℓ > j)

2) train parasite de longueur infinie entraînant une occupation permanente de certains registres, d'où une paralysie supplémentaire du réseau variable selon la cellule (i',j') destinataire du train parasite.

◊ Si le registre Ouest est en panne, le blocage est du même type que pour le registre Est.

Dans ces quatre cas, le blocage du réseau est variable mais généralement il se produit rapidement et totalement (à cause du train parasite).

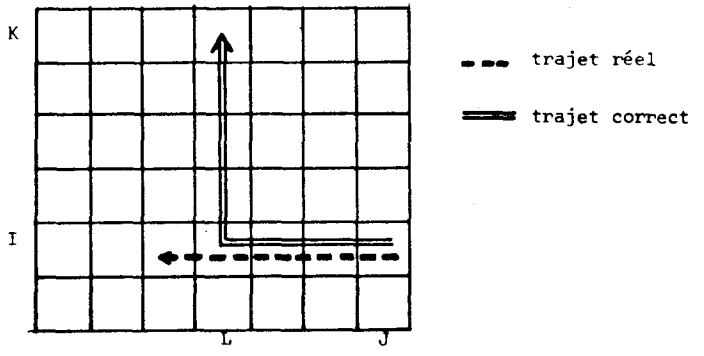
* Le registre M d'une cellule est toujours considéré comme occupé. Dans la mesure où il y a création d'un train parasite de longueur infinie, l'un des cas précédents s'applique.

* Le registre R d'une cellule est toujours considéré comme occupé. Il n'y a pas création d'un train parasite, mais dans ce cas les quatre registres cardinaux se bloquent d'où un blocage généralisé de tout le réseau.

Le blocage du réseau est donc un phénomène rapide et très délicat à contrôler puisque les blocages temporaires d'un train par un autre sont courants. Afin de l'éviter, il faut vraissemblablement contrôler en permanence l'activité du réseau grâce à une unité spécialisée qui peut, en cas de blocage supposé, suspendre rapidement les injections dans le réseau puis activer des procédures (câblées ou plus probablement programmées) de test et de reconfiguration.

IIF3 ERREUR DE ROUTAGE DANS LE RESEAU.

Par suite d'une erreur lors de l'analyse de la locomotive ou parce que la liaison ouverte n'est pas la bonne, un train peut suivre un chemin incorrect.
Exemple :



Le train inséré dans le réseau en (I, J) en direction de (K, L) est mal aiguillé par la cellule (I, L). Le problème du routage est posé à la cellule (I, L-1) car elle ne peut faire parvenir le train en (K, L) par transferts dans le réseau. Il faut donc :

1) qu'une cellule soit capable de détecter un routage impossible

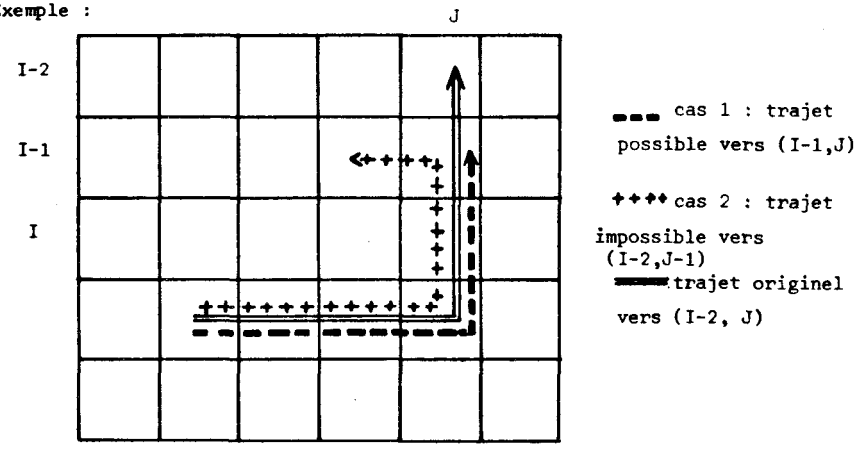
2) qu'une telle erreur soit gérée en dehors du réseau.

IIF4 ALTERATION DU CONTENU D'UN TRAIN.

Cette erreur qui peut être permanente (par exemple à cause d'une position mémoire défaillante) ou fugitive peut affecter soit une locomotive, soit un wagon.

◊ Lorsque la partie adresse destinataire de la locomotive est modifiée, dans certains cas le trajet ainsi défini reste possible (cas 1), dans d'autres le nouveau routage est impossible (cas 2).

Exemple :



L'altération a lieu en (I, J).

Dans le premier cas, l'erreur ne peut être décelée qu'en dehors du réseau, dans le second cas elle doit l'être dans le réseau.

◊ Lorsque le contenu d'un wagon (ou les bits de la locomotive autres que les bits d'adresse) est modifié, le transfert dans le réseau n'en est pas affecté (l'erreur devra donc être décelée hors du réseau) sauf s'il s'agit du bit de suite.

a) Bit de suite à 1 passant à 0.

Il y a création artificielle d'un wagon de queue même si c'est une locomotive, donc fin de train. Le message suivant devra (même si son bit de suite est à zéro) être considéré comme une locomotive, d'où généralement une impossibilité de routage.

b) Bit de suite à 0 passant à 1.

Ce cas ne se produit que sur un wagon de queue. Le registre contenant le message reste "aiguillé" et dès qu'un message y pénètre par la suite,

il est "raccroché" au wagon de queue sur lequel s'est produite l'erreur. A l'inverse du cas précédent (décomposition d'un train), il y a ici concaténation de trains, ce qui n'est décelable qu'en dehors du réseau.

IIF5 GOULOTS CIRCULAIRES DANS LE RESEAU.

Ce sont les situations où un ensemble de messages présents dans le réseau ne peut être acheminé par suite d'un blocage mutuel dû aux ressources utilisées et demandées et donc soit aux ressources de stockage (registres), soit aux ressources de contrôle (logique d'analyse et de transfert). Deux types de goulots peuvent donc se présenter :

a) goulots par registres.

Une telle situation se présente lorsque les demandes forment une liste circulaire, par exemple :

$$(A) \rightarrow B \ ; \ (B) \rightarrow C \ ; \ (C) \rightarrow D \ ; \ (D) \rightarrow A$$

[(A) → B signifie que le contenu du registre A doit aller dans le registre B]. Ceci se traduit par un circuit sur le graphe des liaisons dans le réseau. Cousin a résolu ce type de blocage en imposant le mode de routage (horizontal puis vertical) car ainsi les arcs Nord vers Est, Nord vers Ouest, Sud vers Est, Sud vers Ouest n'existent pas, d'où l'absence de circuits. Malgré la contrainte ainsi posée (le cheminement adaptatif est rendu quasi-impossible), ce mode a été conservé et donc il ne peut exister de circuit causé par les registres.

b) goulots par les ressources de contrôle.

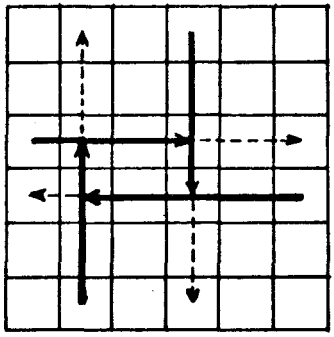
Exemple :

$$(A) \rightarrow E \ ; \ (B) \rightarrow F \ ; \ (C) \rightarrow G \ ; \ (D) \rightarrow H$$

mais le transfert de A vers E est impossible à cause de B, celui de B vers

F à cause de C, etc ...

Cette situation peut être schématisée :



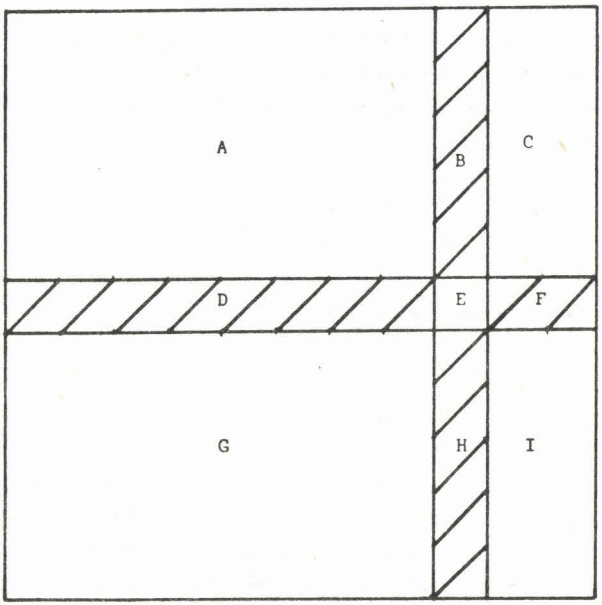
Elle se produit par exemple lorsque la logique de contrôle d'une cellule est séquentielle. Afin de l'éviter, le P.C.C. fonctionne avec une consultation circulaire des registres et un choix aléatoire effectué à chaque cycle (registre de priorités tournantes). Sur le réseau modifié (PROCOPE), la stratégie de l'analyse par une cellule "fournisseur" a pratiquement le même effet.

IIF6 IMPACT D'UNE PANNE.

Lorsqu'une cellule (ou une partie de cellule : registre ou arc originaire d'un registre) est en panne, en plus de la dégradation supportée par les communications avant détection de la panne, il faut évaluer l'impact de la panne i.e. les communications rendues impossibles au niveau du réseau. Ceci revient à déterminer les cas où un train injecté en (i, j), destiné à être extrait en (k, ℓ) doit emprunter la cellule (I, J) fautive. La cellule contient 16 arcs ; il y aura donc 16 possibilités de pannes élémentaires qui sont répertoriées dans le tableau suivant.

Arc	du registre vers le registre	Pas de transfert	
		de	vers
1	N - R	AUBUC	E
2	N - N	AUBUC	H
3	E - S	F	B
4	E - E	F	AUDUG
5	E - R	F	E
6	E - N	F	H
7	S - S	GUHUI	B
8	S - R	GUHUI	E
9	O - S	D	B
10	O - R	D	E
11	O - O	D	CUFUI
12	O - N	D	H
14	M - S	E	B
15	M - E	E	AUDUG
16	M - O	E	CUFUI
17	M - N	E	H

Les zones A, B, C, D, E, F sont délimitées sur le schéma suivant où la zone E correspond à la cellule fautive.



Si un registre est en panne, les transferts impossibles sont donnés par l'ensemble des arcs qui y aboutissent (ou qui en partent) ; par exemple, une panne du registre Sud rend impossibles les trajets de {G ∪ H ∪ D} vers {B ∪ E}.

IIF7 MESURES DE LA FIABILITE DU RESEAU [LEL74].

La fiabilité d'un réseau dépend de caractéristiques matérielles (taux de panne, temps moyen entre pannes) et de caractéristiques structurelles. Ces dernières traduisent la sensibilité du réseau aux pannes, par exemple c'est le nombre d'éléments pouvant tomber en panne sans provoquer une panne totale.

IIF7a Connectivité.

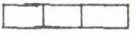
La connectivité en lignes (en noeuds) C d'un réseau est égale au nombre minimum de lignes (de noeuds) devant tomber en panne pour déconnecter deux objets quelconques connectés au réseau. Sur PROCOPE, les cellules sont toutes identiques et l'on a :

$$C_{arcs} = 1 = C_{cellules} = C_{registres}$$

ce qui est faible et justifie la mise en oeuvre systématique de solutions de recours (cf. Procoles de communication).

IIF7b Cohésion.

La cohésion de ligne (de noeud) de degré n, T_n, est définie comme le nombre minimal de lignes (de noeuds) devant tomber en panne pour isoler un ensemble de n noeuds du reste du réseau. Sur le réseau PROCOPE, cette définition n'a de sens que pour des noeuds contigus, mais de plus elle donne des résultats différents selon la position relative des noeuds les uns par rapport aux autres, par exemple :



en cellules : $\Gamma_3 = 8$

$\Gamma_3 = 7$

En considérant les configurations les moins défavorables (les plus groupées) on obtient :

Γ_1 (cellules) = 4 ; Γ_2(cellules) = 6 ; et généralement

Γ_n (cellules) = n+4 $\quad \forall n \geq 1$

Γ_n (lignes) = Γ_n (cellules).

IIF7c Diamètre.

Si d_{ij} est la distance entre 2 noeuds N_i et N_j, le diamètre du réseau est défini par $D = \max_{\{i,j\}} d_{ij}$, d'où

$$D = 2n - 1.$$

Une bonne mesure de la fiabilité est donnée par h (D, D^*) qui représente le nombre minimum de noeuds devant tomber en panne pour que le diamètre du réseau opérationnel passe de D à D^*.
Sur PROCOPE : $D^* = \infty$ d'où $h(D, \infty) = 1$.

IIF3d Survivabilité.

Un noeud (une cellule) est reliée à 4 noeuds voisins ; une cellule a une probabilité de panne p, un arc a une probabilité de panne q (probabilités définies par le matériel). La survivabilité S est définie comme le pourcentage de noeuds pouvant correspondre en permanence avec un noeud pris au hasard. Si le nombre de noeuds tend vers l'infini, ou du moins est grand :

$$S \simeq 1 - \exp(-4 (1 - p) (1 - q) D^*) \Longrightarrow S \simeq 1$$

IIG CONCLUSION.

IIG1 RESUME DES CARACTERISTIQUES DE PROCOPE.

D'après ce qui précède, les principales caractéristiques du réseau de communication PROCOPE sont :

* la transparence : les objets connectés au réseau sont déchargés de la gestion des transferts, il faut toutefois qu'ils gèrent les accès au réseau, celui-ci étant passif

* la modularité : le réseau se compose de cellules identiques, ce qui permet un coût réduit et une maintenance aisée

* l'extensibilité : dans la mesure où les cellules le permettent (champ d'adressage suffisant), il est toujours possible d'ajouter une rangée de cellules au réseau

* la flexibilité : par construction, un objet connecté au réseau peut par celui-ci accéder à tout autre objet connecté

* l'asynchronisme vis-à-vis des objets connectés et le synchronisme interne au réseau

* la simultanéité : la solution de type échangeur simultané (abandonnée pour des raisons de réalisation technologique) aurait permis les échanges simultanés entre cellules voisines. La simultanéité des transferts reste toutefois possible au niveau du réseau ; il est ainsi possible de transmettre en même temps de A vers B et de B vers A

* les performances : étant donnée la rapidité de fonctionnement d'une cellule (de l'ordre de 200 ns par cycle), le volume des échanges est très important

* la souplesse d'utilisation : à l'inverse du P.C.C., les informations échangées ont une longueur indéfinie (trains) ; le rapport $\frac{\text{information utile aux objets connectés}}{\text{information transférée entre cellules}}$ voisin de l'unité en est la preuve

* la fiabilité du réseau est faible ; le réseau n'est utilisable que si la fiabilité peut en être améliorée ou palliée grâce aux objets connectés.

IIG2 CARACTERISTIQUES D'UTILISATION.

En plus des caractéristiques précédentes qui concernent le fonctionnement interne du réseau, il faut noter quelques caractéristiques d'utilisation qui influencent l'architecture machine bâtie sur ce type de réseau :

◊ un objet connecté au réseau peut être sollicité tant en entrée qu'en sortie de façon permanente, il doit donc posséder la capacité de gérer simultanément ces échanges

◊ par construction du réseau, le registre récepteur d'une cellule fonctionne séquentiellement et exclusivement. Un objet connecté ne pouvant être sollicité qu'une seule fois à un instant donné, aucune logique d'arbitrage des accès ne lui est nécessaire.
Inversement, en sortie vers le réseau, le fonctionnement séquentiel et exclusif implique que, si plusieurs objets sont connectés à une même cellule, ils doivent arbitrer leurs conflits mutuels avant d'accéder au réseau. Ce droit d'accès est imprenable tant que le transfert du train en cours n'est pas complet

◊ le réseau est susceptible de blocages rapides, soit temporaires (et localisés) par suite de trains "en collision", soit permanents (en cas de panne). Il est donc nécessaire de vider le réseau le plus rapidement possible, et ce quelle que soit la charge de travail des objets connectés

◊ le délai de transfert est très variable selon les positions relatives des objets concernés ainsi que selon la charge instantanée ; les protocoles de communication doivent donc être très souples. Ceci interdit pratiquement des structures synchrones telles que celle conçue par Flynn [FLY].

◊ un tel réseau ne permet pas l'acheminement simultané de la même information vers plusieurs objets connectés. Cette impossibilité de réaliser un "appel général" interdit des structures telles qu'ILLIAC IV où une unité de contrôle et de commande régit des unités d'exécution [MAC 70].

o les performances du moyen de communication sont tributaires des objets connectés, non seulement en ce qui concerne leurs caractéristiques de débit, mais également en ce qui concerne leur répartition sur l'ensemble du réseau. Cette dernière devra donc être optimisée (cf. Annexe II).

III

PROTOCOLES DE COMMUNICATIONS

IIIA GÉNÉRALITÉS.

Les principales caractéristiques du réseau de transport défini précédemment sont la rapidité, la passivité (cellules câblées sur lesquelles aucun contrôle de flux ni de chemin optimal n'est mis en oeuvre), l'aspect "pipe-line", la simultanéité des transferts.

Ce réseau doit permettre aux objets connectés de se communiquer des informations :

- échanges de données et d'instructions
- transmissions d'ordres
- contrôles d'états logiques et physiques
- enquêtes communes ou partielles sur les états de disponibilité des ressources
- communications collectives d'indications d'intérêt général
- etc...

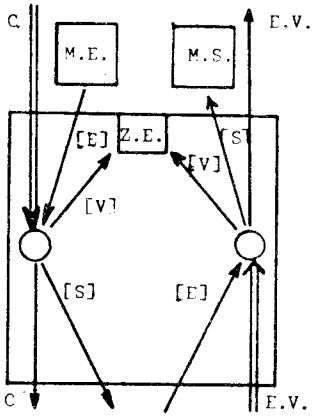
Pour que le système fonctionne de manière satisfaisante et précise, pour éviter des ambiguïtés, des blocages, des pertes d'information, des erreurs de transmission, il importe de définir dans quelles conditions les interlocuteurs peuvent communiquer à travers le réseau.

Dans ce but, une hiérarchie fonctionnelle (i.e. ne tenant pas compte de la réalisation) de machines sera décrite grâce à des notions de réseaux d'ordinateurs (IIIB). Afin de préciser les rôles et relations de ces machines, les parties suivantes décriront les principes généraux régissant les communications (IIIC) puis la mise en oeuvre de ces principes dans le système envisagé (IIID).

IIIB HIÉRARCHIE DE MACHINES.

IIIB1 MODELE DE MACHINE [LEL74]

Une machine peut généralement être décrite selon le schéma suivant :



==> : stimulations provenant de l'extérieur de la machine

→ : conséquences possibles d'une stimulation

C : commande

E.V. : événement

[E] : instruction d'entrée

[S] : instruction de sortie

[V] : visualisation d'état

M.E. : mémoire d'entrée

M.S. : mémoire de sortie

Z.E. : zone de visualisation d'état.

Fonctionnellement, une machine M est un automate capable de prendre plusieurs états ou d'exécuter des instructions. L'automate est dirigé de l'extérieur, soit par des commandes issues de la machine utilisant M, soit par des événements issus de la machine utilisée par M. L'automate dispose d'un jeu d'instructions lui permettant d'effectuer des émissions de commandes ou d'événements, des opérations d'entrée et de sortie, des visualisations d'état. Pour cela, il dispose d'une zone d'état (Z.E) observable par la machine utilisatrice, ainsi que de mémoires d'entrée (M.E.) et de sortie (M.S.) partagées avec la machine utilisatrice. Ce modèle permet de décrire les différentes machines fonctionnelles du système en tenant compte d'une part de la répartition géographique d'une machine, d'autre part de la décomposition de chaque machine en composants.

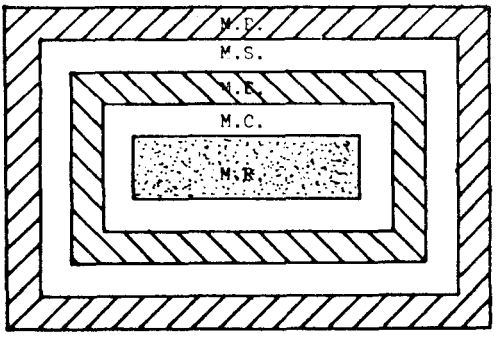
Ce modèle est récursif, c'est-à-dire que c'est également celui d'un composant de machine ; il permet donc de représenter une machine par juxtaposition de plusieurs composants identiques et de représenter le système comme l'empilement de plusieurs couches de machines.

IIIB2 NIVEAUX DE MACHINES.

Dans le système étudié, nous avons distingué cinq niveaux de machines :

- machine de commutation (ou machine réseau) : M.R.
- machine de communication : M.C.
- machine d'échanges : M.E.
- machine de services : M.S.
- machine processus : M.P.

Ces cinq niveaux sont hiérarchisés par la relation d'utilisation, e.g. la machine processus utilise la machine de services pour satisfaire ses besoins d'échanges, la machine de services utilise la machine d'échanges, etc...



Voici une description succinte de leurs rôles respectifs qui seront détaillés en IIID.

IIIB2a La machine de commutation (M.R.)

C'est le réseau de transport tel qu'il a été défini précédemment. La machine M.R. est donc formée par juxtaposition des n^2 machines élémentaires que sont les *cellules*, automates câblés ne disposant que d'opérations d'entrée, de sortie et de visualisation d'état. La machine de commutation manipule des *éléments*. Ces éléments sont groupés en *trains* dont le premier élément appelé locomotive définit le point de sortie hors de la machine M.R.

IIIB2b La machine de communication (M.C.)

Elle est formée de *stations de communication* (S.C.) ; une station de communication est associée à chaque cellule. La machine M.C. soumet des *éléments* à la machine de commutation ; elle a pour principaux rôles :

- la gestion du routage
- l'insertion et l'extraction des éléments
- l'adaptation des vitesses avec la machine de commutation
- l'exécution d'ordres pour la gestion du routage
- la détection d'erreurs de routage.

IIIB2c La machine d'échanges (M.E.)

Elle est formée de *stations d'échanges* (S.E.) ; une station d'échanges est associée à chaque station de communication. Elle soumet (et reçoit) des séquences de *messages* vers (de) la machine de communication ; ses principaux rôles sont :

- contrôles d'erreur sur le contenu d'une séquence
- mise en forme
- gestion des communications ne faisant pas appel à la M.C.
- tri et aiguillage vers les stations de services
- gestion de l'adressage des unités connues par un nom générique (cf. IIID5b).

IIIB2d La machine de services (M.S.)

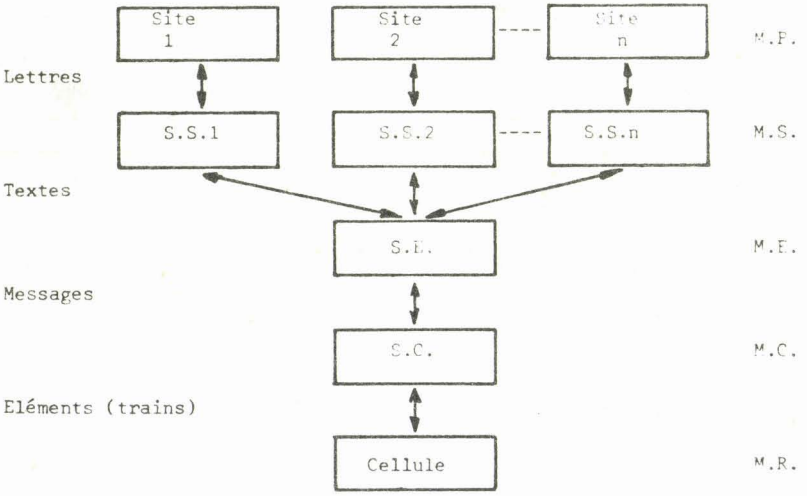
Elle est formée de *stations de services* ; une station de services est associée à chaque composant de la machine processus dont elle reçoit (et vers laquelle elle envoie) des *textes*. Ses principaux rôles sont :

- la gestion des protocoles de la machine processus
- la gestion de l'allocation (cf. IIID5b).
- la gestion des types de textes (priorité éventuelle, interrogations, délégations et réponses).

IIIB2e La machine processus (M.P.)

C'est l'ensemble des fonctions de traitement et de gestion du traitement s'exécutant surles ressources du système. Elle comprend donc en particulier tous les processeurs. Un composant de la machine processus, appelé *site*, manipule des *lettres*.

Cette hiérarchie peut se schématiser :



IIIB3 PROTOCOLES.

Chaque machine est chargée de missions qu'elle accomplit en utilisant la machine de niveau inférieur. Ces missions lui sont notifiées soit par la machine de niveau immédiatement supérieur, soit par la machine processus. Les relations de coopération et de communication existant dans le système sont :

- horizontales à l'intérieur d'une même machine
- verticales entre machines différentes.

Ces relations obéissent à des lois : les protocoles, et sont mises en oeuvre par des primitives. Les protocoles se subdivisent donc en protocoles internes (relations horizontales) et protocoles externes (relations verticales). Les fonctions d'un protocole interne sont :

- le nommage : description des espaces de noms utilisés
- le transfert qui se décompose en :
  - la transformation descendante du format des données à la commande (vers la machine de niveau inférieur)
  - le transfert des données à travers la machine de niveau inférieur (utilisation de cette machine selon son protocole)
  - la transformation ascendante du format des données à la livraison (en provenance de la machine de niveau inférieur).

Ce transfert ne peut jamais être totalement fiable et deux fonctions annexes lui sont généralement associées :

- contrôle d'erreur
- contrôle de flux afin d'éviter de saturer tout composant d'une machine.

Les fonctions d'un protocole externe sont :

- la commande : description des commandes et données reçues en M.E.
- la livraison : description des événements postés, des données inscrites en M.S., des états visualisés en Z.E. destinés à la machine de niveau supérieur
- contrôle de flux externe : conditionne l'utilisation d'une machine par ses usagers
- contrôle d'erreur externe : rend compte à un usager de la façon dont s'est exécutée une commande.

IIIC PRINCIPES GÉNÉRAUX RÉGISSANT LES COMMUNICATIONS DANS UN SYSTÈME MULTIPROCESSEUR.

IIIC1 INTRODUCTION.

Les protocoles ont été définis comme les lois régissant les relations de coopération et de communication existant dans le système. Cette section traite essentiellement des règles de construction d'échanges corrects d'information entre composants de la machine processus, ce qui ne recouvre qu'une partie du domaine des protocoles défini en IIIB3. La notion de protocoles est associée aux réseaux d'ordinateurs où ils furent en premier mis en oeuvre. Ils s'y caractérisent essentiellement :

- d'un point de vue qualitatif :
  - le moyen de communication est un support de télécommunications (lignes P & T, faisceaux hertziens)
  - les composants coopèrent pour l'acheminement des informations et la gestion y afférant, rarement pour les traitements
  - les échanges ont une structure rudimentaire puisqu'il s'agit souvent d'un ensemble de dialogues (2 interlocuteurs concernés).

- d'un point de vue quantitatif :
  - les distances de transmission sont importantes (jusque plusieurs milliers de km)
  - les volumes d'information échangés le sont soit par quanta de taille importante, soit par quantités transférées variables mais généralement élevées
  - les composants concernés par les échanges (généralement des ordinateurs) sont de grande puissance de traitement, de stockage, ils comprennent pratiquement toujours une unité de contrôle et de commande évoluée et performante

- les composants acheminant les informations sont eux-mêmes des (minis) ordinateurs aux performances non négligeables.

Ces caractéristiques ne sont pas sans influer lourdement sur la mise en oeuvre des protocoles dont les réseaux ARPANET [KLE76], CYCLADES [LEL74] offrent des exemples complexes.

Des réseaux tels qu'ETHERNET [MET76] sont proches de ces réseaux d'ordinateurs par certains aspects (transmission par câble coaxial, structure de dialogues) mais s'en éloignent sur d'autres (distance de l'ordre du kilomètre, composants de puissance limitée puisqu'il s'agit de mini-ordinateurs, volumes d'information échangés de taille plus faible que sur les réseaux d'ordinateurs, coopération entre unités envisageable au niveau de certains traitements).

Les systèmes multiprocesseurs sont également candidats à la mise en oeuvre de protocoles de communication, particulièrement les machines réseaux dont la structure avoisine celle de certains réseaux d'ordinateurs. Dans ces systèmes, les protocoles se caractérisent plutôt comme suit.

D'un point de vue qualitatif :

- le moyen de communication est de nature très variable (réseau à décalage, bus, réseau cellulaire ...) entre systèmes, voire à l'intérieur d'un même système (hétérogénéité)
- les composants du système doivent non seulement co-opérer pour l'acheminement des informations mais également pour les traitements
- parallèlement à cette coopération, les échanges doivent utiliser des structures plus complexes mettant en jeu un nombre quelconque d'unités et un haut niveau de parallélisme. Ces structures se rapprochent des structures des "conversations" humaines directes ou par l'intermédiaire de réseaux informatiques [VAL76].

0 D'un point de vue quantitatif :

- les distances de transmission sont faibles
- les volumes d'information échangés sont très variables selon les composants observés, mais généralement un transfert est de taille réduite par rapport aux réseaux d'ordinateurs.
- les composants transmettant l'information sont :
  - de puissance limitée
  - de possibilités diverses (mémoires, processeurs ...) ; en particulier certains peuvent ne pas posséder d'unité de contrôle et de commande.
- les composants acheminant les informations sont soit des processeurs aux possibilités limitées, soit des organes câblés au fonctionnement simple.

La mise en oeuvre de tels protocoles a été peu étudiée et les tentatives de réalisation sont très diverses. Citons par exemple : l'ILLIAC IV [MAC70] (pas de protocoles d'échanges), le réseau de Koczela [KOC69] (protocoles rudimentaires reposant essentiellement sur l'état des composants et le mode d'échange par noms ou par niveaux).

Les concepts généraux développés dans cette section tentent de définir des protocoles satisfaisant aux caractéristiques précédentes et relativement indépendants du moyen de communication utilisé.

Dans cette section, le message (notion floue car une définition précise dépend de la technologie de transmission, de l'application envisagée, ...) est défini comme l'ensemble des informations susceptibles d'être transférées au cours d'une phase élémentaire de travail des composants qui communiquent. Ces composants seront également appelés usagers, indépendamment de leur nature, de leur réalisation (matérielle ou logicielle), voire du niveau de machine concerné, afin de définir une structure fonctionnelle homogène indépendante de la mise en oeuvre.

IIIC2 NOTION INTUITIVE DE PROTOCOLES.

Un usager A chargé d'un travail communique avec d'autres usagers dans divers cas dont :

a) lorsqu'il ne peut seul faire progresser ce travail. Dans ce cas il sollicite un autre composant afin de l'aider ; il devra évidemment être informé des suites données à sa demande (par exemple une demande de ressource logicielle ou matérielle)

b) lorsqu'il a été sollicité et répond à cette sollicitation

c) lorsque son travail lui impose de fournir une information à un autre usager. Dans ce cas, il peut vouloir en vérifier la transmission et l'utilisation

d) lorsqu'il a reçu une information qu'il s'est appropriée.

Par la suite, nous supposerons qu'il existe explicitement, pour tout échange et pour tous les usagers impliqués dans cet échange, un début (avant lequel les usagers ont des fonctionnements indépendants) et une fin (après laquelle les fonctionnements sont de nouveau indépendants). Ceci implique qu'un usager qui en informe un autre en attende un écho. Les échanges définis ci-dessus amènent la mise en oeuvre de trois mécanismes de base :

* *interroger* : i.e. créer puis transmettre une demande à un autre usager

* *répondre* : i.e. élaborer puis transmettre une information satisfaisant une demande antérieure ; cette information est en principe destinée à l'usager interrogateur

* *déléguer* : i.e. ne pouvoir satisfaire une demande mais la transmettre à un autre usager susceptible de le faire.

Selon ces mécanismes, il existe deux types de messages :

* *des requêtes* (M_r) dont le contenu demande l'utilisation d'une ressource externe à l'usager qui les a émises

* *des services* (M_s) dont le contenu apporte à un usager une ressource dont il avait précédemment requis l'intervention.

Il se peut qu'un usager en interroge plusieurs autres d'affilée sans en attendre de réponse immédiate ; de même, à un instant donné, un usager peut être sollicité par plusieurs autres et traiter les différentes requêtes au fur et à mesure de sa disponibilité. En conséquence, un usager interrogateur doit pouvoir mémoriser sa requête et la relier à l'environnement dans lequel elle a été créee (par exemple, le mot d'état du processus à cet instant), reconnaître et affecter correctement les services reçus, effectuer divers contrôles, ré-itérer une requête lorsque l'attente du service est estimée trop grande ou lorsque le service reçu n'est pas satisfaisant, etc...

De même, un usager recevant une requête (suite à une interrogation ou à une délégation) doit pouvoir mémoriser différentes requêtes, distribuer correctement les suites (réponses ou délégations) qu'il y donne, effectuer divers contrôles, etc...

En plus de la relation requête-environnement, du contenu de la requête, un usager interrogateur doit pouvoir mémoriser une identité de la requête : l'*identificateur*. D'autre part, la requête doit être accompagnée d'une identité la repérant de façon unique dans tout le système ; cette identité doit également repérer les suites données à cette requête.

Ces définitions intuitives appellent deux remarques principales :

a) la notion de délégation ne permet pas de décrire certains types d'échanges, en particulier ceux où un usager ne satisfait qu'une partie de la requête (par exemple les travaux "pipe-line")

b) les relations entre mécanismes de base et identificateurs ne sont pas définies.

Les mécanismes de base doivent donc être définis plus précisément, en particulier la délégation ; de même pour le rôle et la "vie" des identificateurs.

IIIC3 DEFINITION DES MECANISMES DE BASE.

IIIC3a Messages.

Un message comprend trois parties :

a) origine : elle précise d'une part la référence d'un usager (l'émetteur) d'autre part un identificateur. Ce couple doit permettre le repérage du message dans tout le système

b) contenu (M) : deux types de messages (requête, service) ont été définis précédemment selon leur contenu. Afin de généraliser la notion de délégation, nous dirons qu'un message contient dans des proportions variables une partie requête et une partie service ($M = M_r \,||\, M_s$).

c) destinataires : cette partie indique d'une part explicitement la référence de l'usager auquel est destiné le message transmis ; d'autre part elle spécifie sous une forme très variable selon les réalisations (liste explicite, fonction de détermination, etc...) des références d'usagers qui manipuleront éventuellement le message à des stades ultérieurs de son évolution.

IIIC3b Interroger.

Définition.

Un usager A interroge un usager B lorsqu'il requiert l'intervention d'une ressource disponible (ou supposée telle) par B afin de faire progresser son travail A.

Notation.

$$A \overset{I}{\dashv} B$$

0 La référence d'usager (émetteur) de la zone origine est celle de l'usager qui interroge ; celui-ci crée également l'identificateur et la zone "destinataires".

0 Le message de A vers B contiendra soit une requête pure, soit une partie service (informations connues de A à utiliser par B) et une partie requête. La requête pourra être complexe, voire composée d'un ensemble de requêtes plus élémentaires (par exemple un ensemble de références mémoires).

0 *Lorsqu'un usager A interroge un usager B, il crée un identificateur qu'il mémorise et qu'il insère dans le message émis vers B.*

L'usager B interrogé mémorisera le message reçu, y compris la partie origine et la partie destinataires.

0 La capacité d'un usager d'en interroger d'autres est une caractéristique importante pour définir une hiérarchie des usagers (cf.IIIC7)

Exemple.

Soit un usager A (un processeur) qui doit trouver parmi un ensemble de mémoires associatives (S_1, S_2, ... , S_n) trois chaînes de caractères commençant par la lettre B.

L'usager A crée un identificateur Id_o et le message émis sera par exemple :

$A \overset{I}{\rightarrow} S_1$

Origine { A ; Id_o

Contenu { M_r = "SEARCH (B,3)" ; $M_s = \emptyset$

Destinataires S_n, S_{n-1},...,S_1.

(S_1 : destinataire du message transmis)

IIIC3c Déléguer.

Suite à une interrogation qu'il a reçue, un usager tente d'apporter sa contribution au travail demandé soit seul, soit en interrogeant d'autres usagers.

Cette contribution peut selon les cas être totale, nulle ou partielle.

a) Elle est totale lorsque l'usager élabore un message ne contenant qu'une partie service ; dans ce cas l'usager répond (le refus et la déclaration d'indisponibilité temporaire explicites sont des réponses).

b) Elle est nulle lorsque l'usager ne peut que transmettre le message à un autre usager susceptible d'y travailler (notion précédente de délégation). Dans ce cas, comme dans le suivant, la zone "destinataires" du message reçu permet de définir le destinataire du message transmis.

c) Elle est partielle lorsque l'usager a pu réduire sans l'annuler la partie requête et a pu augmenter la partie service. Cette situation correspond à un enrichissement progressif (travaux "pipe-line") et généralise la définition de la délégation.

Définition.

Un usager A qui a reçu un message contenant une partie requête non vide délègue vers un usager B lorsqu'il lui transmet un message comprenant une partie requête non vide, identique ou moins importante que celle du message reçu.

En conséquence, la partie service sera identique ou plus importante que celle du message reçu.

Notation.

$$A \overset{D}{\rightarrow} B$$

Lorsqu'un usager A délègue vers un usager B, la zone origine du message transmis est identique à celle du message incident correspondant ; une fois le message transmis, l'usager A ne garde aucune trace de ces messages.

L'usager B se comporte comme s'il était interrogé.

Exemple. (Suite de l'exemple précédent)

$$A \underset{(1)}{\overset{I}{\rightarrow}} S_1 \underset{(2)}{\overset{D}{\rightarrow}} S_2 \underset{(3)}{\overset{D}{\rightarrow}} S_3 \underset{(4)}{\overset{D}{\rightarrow}} S_4 \ \ldots \ S_k \qquad (k \leq n)$$

Description des messages :

(1) : cf. IIIC3b

(2) :

- **origine** : A, Id_o
- **contenu** : M_{2r} = "SEARCH (B,2)", M_{2s} = "BONBON"
- **destinataires** : $S_n, S_{n-1}, \ldots, \underline{S_2}$

(3) :

- **origine** : A, Id_o
- **contenu** : M_{3r} = "SEARCH (B,2)", M_{3s} = "BONBON"
- **destinataires** : $S_n, S_{n-1}, \ldots, \underline{S_3}$

(4) :

- **origine** : A, Id_o
- **contenu** : M_{4r} = "SEARCH (B,1)", M_{4s} = "BONBON, BONJOUR"
- **destinataires** : $S_n, \ldots, \underline{S_4}$

Remarques :

a) cet exemple ne présente qu'une solution particulière possible où un seul résultat au plus est trouvé dans chaque mémoire

b) si la relation $M_r > M'_r$ signifie que la requête de M_r est plus importante que celle de M'_r et si $M_r = M'_r$ signifie qu'elles sont identiques, il vient :

$M_{1r} > M_{2r}$ et $M_{2s} > M_{1s}$ $\Rightarrow$ S_1 a contribué partiellement au travail

$M_{3r} = M_{2r}$ et $M_{3s} = M_{2s}$ $\Rightarrow$ S_2 n'a pas contribué à l'enrichissement du travail

$M_{3r} > M_{4r}$ et $M_{4s} > M_{3s}$ $\Rightarrow$ S_3 a contribué partiellement au travail.

IIIC3d **Répondre**.

Définition.

Un usager A *répond à un usager* B *lorsque le contenu du message transmis ne contient qu'une partie service.*

(la partie requête est vide).

Notation.

$$A \overset{R}{\downarrow} B$$

D'après ce qui précède, l'usager qui reçoit une réponse doit être celui qui a crée l'interrogation correspondante (mêmes zones origines dans les messages).

Lorsqu'un usager A répond à un usager B, il se comporte vis-à-vis de la zone origine comme pour une délégation.

Un usager qui reçoit une réponse doit, après avoir utilisé le service transmis, détruire le message reçu ainsi que l'identificateur stocké correspondant.

Exemple (suite de l'exemple précédent).

$$S_k \overset{R}{(5)} A$$

Contenu du message : (5)

origine { A ; Id_o

contenu { $M_r = \emptyset$; M_s = "BONBON, BONJOUR, BONHOMME"

destinataire A

III.4 ASSOCIATIONS DE MECANISMES DE BASE.

Partant des mécanismes de base nous définissons ici leurs associations permettant d'organiser un ensemble d'échanges entre des usagers qui coopèrent afin de faire progresser un travail. Cet ensemble d'échanges forme une *conférence*. Cette notion décrit une organisation fonctionnelle (par exemple les dépendances logiques) qui n'est pas nécessairement isomorphe à la structure des échanges dans le moyen de communication, celle-ci

tenant compte du déroulement dans le temps et dans l'espace (routage).

Lorsqu'un usager A interroge un usager B, cela peut entraîner des échanges entre usagers qui se terminent par un message de réponse transmis à l'usager A. Cet ensemble d'échanges suivant l'interrogation est appelé *tâche de B pour A* et noté $B \overset{T}{\dashv} A$. La *consultation de A*, notée $A \overset{C}{\dashv} A$, est la donnée d'une interrogation de A vers B et de la tâche de B pour A qui s'en suit. D'où :

P1 $$A \overset{C}{\dashv} A = A \overset{I}{\dashv} B \overset{T}{\dashv} A$$

Un *colloque* de A, noté $A \overset{\mathcal{C}}{\dashv} A$, est un ensemble fini de consultations de A ne présentant pas de dépendances logiques entre elles. D'où :

P2 $$A \overset{\mathcal{C}}{\dashv} A = \left\{ \begin{array}{l} A \overset{C_1}{\dashv} A \\ \text{------} \\ \text{------} \\ A \overset{C_n}{\dashv} A \end{array} \right\}$$

Cette notion de colloque permet donc d'introduire le parallélisme entre consultations ; certes la technologie impose généralement que les diverses interrogations placées au début des consultations d'un même usager ne puissent être envoyées simultanément dans le moyen de communication ; ce parallélisme est essentiellement fonctionnel au niveau de l'usager A et traduit le fait qu'il ne doit pas attendre la conclusion de la consultation C_i avant de générer l'interrogation débutant C_{i+1}, l'ordre des C_i étant par ailleurs indifférent.

Un colloque $A \overset{\mathcal{C}}{\dashv} A$ peut toujours se mettre sous la forme d'une suite de colloques :

$$A \overset{\mathcal{C}_1}{\dashv} A \overset{\mathcal{C}_2}{\dashv} A \ \ldots \ A \overset{\mathcal{C}_k}{\dashv} A$$

(où un colloque peut se réduire à une consultation).

En revanche, la suite de colloques :

$$A \overset{\mathcal{C}'_1}{\rightarrow} A \overset{\mathcal{C}'_2}{\rightarrow} A \overset{\mathcal{C}'_3}{\rightarrow} A \ \ldots \ A \overset{\mathcal{C}'_k}{\rightarrow} A$$

ne peut généralement pas être représentée par un seul colloque $A \overset{\mathcal{C}_1}{\rightarrow} A$ à cause des dépendances logiques entre les $\mathcal{C}'_j$.

Définition de la tâche de B pour A.

Ce peut tout d'abord être l'élaboration immédiate d'une réponse de B vers A :

$$P3 \qquad B \overset{T}{\dashrightarrow} A = B \overset{R}{\rightarrow} A$$

Ce peut être la délégation de B vers E suivie d'une tâche de E pour A :

$$P4 \qquad B \overset{T}{\dashrightarrow} A = B \overset{D}{\rightarrow} E \overset{T}{\dashrightarrow} A$$

Enfin, avant de répondre à l'usager A, l'usager B peut réclamer à d'autre usagers des informations complémentaires. Il entreprend donc des consultations, ou plus généralement des colloques. D'où :

$$P5 \qquad B \overset{T}{\dashrightarrow} A = B \overset{\mathcal{C}}{\rightarrow} B \overset{T}{\dashrightarrow} A$$

Des règles précédentes ainsi que des définitions des mécanismes de base nous déduisons la forme générale d'une tâche de B pour A :

$$B \overset{T}{\dashrightarrow} A = B_o \overset{\alpha_1}{\rightarrow} B_1 \overset{\alpha_2}{\rightarrow} B_2 \rightarrow B_3 \ \ldots \ B_n \overset{R}{\rightarrow} A$$

où : $\alpha_i = D$; $B_i \neq B_{i-1}$ ou bien $\alpha_i = \mathcal{C}$; $B_i = B_{i-1}$

ce qui peut s'écrire plus concisément : $T = (D + \mathcal{C})^* R$

Une consultation de A est alors de la forme :

$$A \xrightarrow{C} A = A \xrightarrow{I} B_o \xrightarrow{\alpha_1} B_1 \xrightarrow{\alpha_2} B_2 \ldots B_n \xrightarrow{R} A$$

ce qui s'écrira plus simplement

$$C = I(D + \mathcal{C})^* R$$

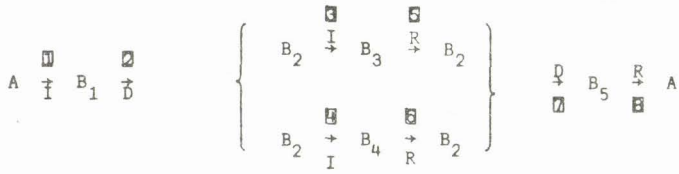
Remarques.

1) Ces définitions permettent certaines situations particulières telles que

$$A \xrightarrow{I} B \xrightarrow{\mathcal{C}} B \xrightarrow{D} E$$

où le message délégué par B vers E a le même contenu que celui reçu par B de A. Cela correspond à une tentative de contribution par B qui a échoué.

2) Exemple de conférence



Contenu des différents messages :

n°	Partie requête	Partie service
1	valeur de 1+2+3+4 ?	ø
2	valeur de 1+2+3+4 ?	ø
3	valeur de 1+2 ?	ø
4	valeur de 3+4 ?	ø
5	ø	3
6	ø	7
7	valeur de 3+7 ?	ø
8	ø	10

IIIC5 IDENTIFICATEURS.

Un message comprend une partie origine, une partie contenu et une partie destinataires. La partie origine est un couple :

- référence de l'usager qui a crée l'interrogation débutant la consultation (spécification spatiale)
- identificateur (spécification ordinale).

Ce couple doit permettre le repérage de la consultation (et du message) dans tout le système ; l'identificateur est crée par un usager indépendamment des autres usagers. La référence doit donc être unique et se compose généralement :

- de l'adresse du point d'accès de l'usager au moyen de communication
- d'un numéro d'ordre repérant l'usager parmi ceux connectés à ce point d'accès.

Etant donné les délais de transmission, les erreurs possibles, etc... la référence doit être non seulement unique mais permanente. En revanche, l'identificateur ne doit qu'être unique à tout instant (i.e. une seule consultation de référence donnée au plus peut avoir un identificateur donné).

Définition.

Un identificateur est une marque repérant de façon unique la consultation à laquelle il est associé parmi toutes celles qui, à un instant donné, ont pour origine l'usager qui l'a crée.

Exemple :

un identificateur peut être obtenu par concaténation du nom du processus cause du message et d'une date de création.

IIIC5a Vie d'un identificateur.

D'après ce qui précède rappelons que :

- un identificateur est crée par un usager lors d'une interrogation
- il est inséré dans le message et stocké par l'usager
- il n'est pas modifié par délégation ou réponse mais transmis
- il est détruit par l'usager interrogateur après réception de la réponse correspondante.

IIIC5b Données annexes à un identificateur stocké par un usager interrogateur.

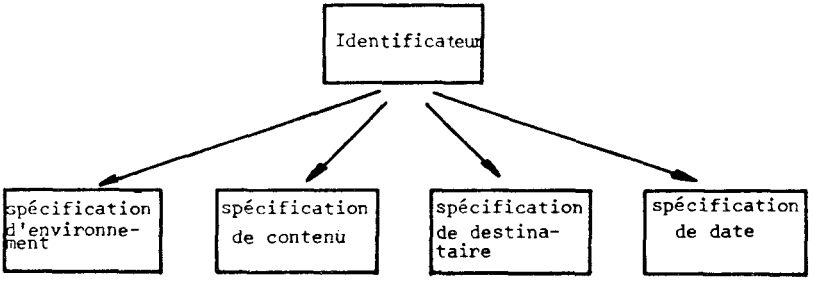
Un identificateur est stocké par l'usager interrogateur ; diverses informations lui sont associées :

◊ l'environnement de l'usager lors de l'interrogation (cf. IIIC2). Cette information est superflue si l'usager ne peut changer d'environnement entre l'interrogation et la réception de la réponse, ni de lui-même, ni du fait d'une excitation externe (réception d'une interrogation, d'une délégation...). Dans d'autres cas elle est plus ou moins complexe (par exemple si l'usager est un processeur fonctionnant en multiprogrammation).

◊ contenu du message (cf. IIIC2)

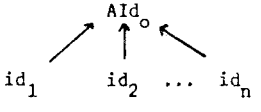
◊ la partie "destinataires" du message. Lorsque le contenu est à ré-utiliser, il peut être utile de disposer de cette liste, soit pour la recopier, soit pour essayer d'autres destinataires (s'ils existent).

◊ diverses informations d'exploitation, dont une date de création (si elle n'est pas intégrée à l'identificateur) peuvent être nécessaires, par exemple afin d'armer un mécanisme du type "chien de garde" ou tout autre mécanisme de réveil entraînant une répétition de l'interrogation après un temps d'attente estimé trop long.



Identificateurs et conférences.

Soit un usager B chargé d'une tâche pour un usager A ; il a donc reçu précédemment un message M de partie origine (A, id_o) et de contenu requête Q. Cette tâche de B pour A peut se poursuivre soit par une réponse de B à A, soit par une délégation de B vers C, soit par une suite de colloques de B suivie d'une tâche de B pour A. Dans les deux premiers cas, la gestion des identificateurs a été décrite ; nous nous intéressons ici au troisième cas où B doit créer de nouveaux identificateur pour chacune des consultations qu'il débute. Afin de gérer correctement le message (A, id_o), l'usager B doit établir des relations entre les identificateurs (id_1, id_2 ... id_n) des consultations émises, et le couple (A, Id_o). Ainsi B pourra reprendre le traitement du message M après avoir reçu les services requis dans le colloque.



Ces relations doivent permettre une grande souplesse d'utilisation car il se peut, par exemple, que l'usager B puisse reprendre le traitement avant d'avoir reçu les réponses à tous les messages interrogateurs envoyés. Certaines réponses pourront même se révéler superflues relativement aux autres.

Une solution à ce problème peut être que B garde en mémoire les identificateurs non encore détruits ; il les détruira à la réception des réponses correspondantes. Les liens des consultations superflues avec $(A, I\beta_o)$ seront remplacés par une marque "inutile". La réception d'une réponse marquée "inutile" n'entraînera que la destruction de l'identificateur et des informations stockés. Toutefois, dans le cas où les services de ces réponses inutiles sont des avis de réservation de ressources, il faudra les libérer grâce à une nouvelle consultation de B.

IIIC6 EXEMPLES ET CAS PARTICULIERS.

IIIC6a Application au calcul d'une expression arithmétique.

Partant d'un exemple très simple, cette section montre comment la tâche consistant à calculer la valeur d'une expression arithmétique peut être effectuée par un ensemble d'usagers utilisant les protocoles définis. Après une définition des usagers et des messages seront explicités les moyens de gestion des protocoles par un usager et les méthodes correspondantes. Le moyen de communication est supposé idéal (i.e. sans interférence sur le fonctionnement des usagers). Les expressions arithmétiques envisagées seront, par souci de simplicité, formées uniquement de l'opération d'addition et de nombres. Une généralisation sera amorcée vers des expressions plus complexes et variées.

Les messages.

Soit un ensemble d'usagers $A = \{A_1, A_2, \ldots, A_p\}$. Tous ces usagers sont capables d'effectuer des additions ainsi que les travaux de contrôle et de gestion des protocoles. Un *terme* est défini inductivement par :

- tout nombre est un terme
- si t et t' sont des termes, +(t,t') est un terme.

Cette définition des termes ne précise rien quant à leur représentation (polonaise pré ou post-fixée, infixée totalement parenthésée ou non ...). Généralement, afin de tenir compte des extensions possibles, un message sera un quintuplet :

$$<A_i, Id, t, \mathcal{D}, A_j>$$

A_i : référence de l'usager interrogateur

Id : identificateur

t : terme

$\mathcal{D}$: partie de A ; c'est l'ensemble des destinataires potentiels (i.e. les usagers vers lesquels le message pourra être délégué s'il n'est pas pris en compte par le destinataire effectif A_j)

A_j : référence d'un usager faisant partie de $\mathcal{D}$; c'est l'usager auquel est destiné le message considéré.

Dans le cas simple considéré, deux types de messages sont nécessaires ; ils sont parfaitement différentiables :

* *les questions*

Ce sont les messages où :

- t n'est pas un nombre
- $\mathcal{D} = A$ i.e. tous les usagers sont destinataires potentiels ; ils forment une liste explicite dans le message

* *Les réponses*

Ce sont les messages où :

- t est un nombre
- $\mathcal{D} = A_i$ (l'usager interrogateur est le seul destinataire potentiel) (==> $A_j = A_i$)

Tables d'identificateurs.

A chaque usager A_i sont associées deux tables R_i et C_i.

a) Table R_i.

C'est la table des identificateurs reçus, formée d'éléments tels que

$$<A_j, Id, n, X, Y>$$

Chaque élément vérifie :

C_1 I) X et Y sont soit des nombres, soit des identificateurs et l'un au moins est un identificateur

II) n est égal au nombre d'identificateurs dans le couple (X, Y). Donc $n \in \{1,2\}$.

b) Table C_i.

C'est la table des identificateurs crées, formée d'éléments tels que

$$<Id, A_j, Id'>$$

Chaque élément vérifie :

C_2 tous les identificateurs qui figurent en première composante des éléments de C_i à un instant donné sont différents

Enfin, les tables R_i et C_i doivent vérifier la condition suivante.

C_3 i) si $\langle A_j, Id, n, X, Y\rangle \in R_i$ et $Id' \in (X,Y)$ alors $\langle Id', A_j, Id\rangle \in C_i$

ii) si $\langle Id', A_j, Id\rangle \in C_i$, il existe $\langle A_j, Id, n, X, Y\rangle \in R_i$ tel que $Id' \in (X,Y)$.

On verra ci-dessous dans la description du fonctionnement du système que lorsque des identificateurs sont crées par un usager A_i, ils le sont lors de la création de tâches de niveau inférieur afin de pouvoir répondre à une question reçue par cet usager.

La table C_i contient donc les identificateurs crées ainsi que la référence de la question qui a causé leur création (condition C_3 (ii)).

La table R_i contient les références (zones origine) des questions reçues et les identificateurs dont elles ont provoqué la création (condition C_3(i)).

Action des usagers.

Que font les usagers lorsqu'ils reçoivent un message (question ou réponse) ?

a) Réception d'une question.

Tout d'abord, lorsqu'un usager reçoit une question, il peut décider pour une raison quelconque, de la transmettre purement et simplement à un autre usager. Pour cela il suffit de fabriquer un nouveau message ne différant du précédent que par la référence du destinataire effectif.

Si l'usager A_j accepte de prendre en compte la question $\langle A_i, Id, t, A, A_j\rangle$ qui lui est adressée, par définition le terme t est de la forme $+(t_1, t_2)$.

1er cas : t_1 et t_2 sont tous deux des nombres. A_j calcule le résultat $x = t_1 + t_2$ et envoie la réponse $<A_i, I_d, x, \{A_i\}, A_i>$. Les tables R_j et C_j sont les mêmes qu'avant réception du message (après un éventuel stockage temporaire).

2ème cas : t_1 et t_2 ne sont pas deux nombres. Supposons que t_1 ne soit pas un nombre mais que t_2 le soit (les autres cas seraient traités de façon similaire).

L'usager A_j crée un nouvel identificateur Id' et ajoute à la table C_i l'élément $<Id', A_i, Id>$; il ajoute à la table R_i l'élément $<A_i, Id, 1, Id', t_2>$ puis envoie la question $<A_j, Id', t_1, A, A_k>$. Les conditions C_1 à C_3 sont toujours vérifiées par les tables R_j et C_j tandis que l'usager attend sa réponse.

b) Réception d'une réponse.

Si l'usager A_i reçoit le message $<A_i, Id, x, \{A_i\}, A_i>$, c'est une réponse ($\mathcal{D} = \{A_i\}$) et le message ne peut être transmis Cette réponse correspond à une question posée par A_i ; soit que cette question était la question initiale posée par A_i et la consultation est achevée, soit que cette question ait été posée par A_i pour pouvoir répondre à une autre question (pouvant à la limite être une excitation d'origine externe). La table C_i contient alors un unique élément $<Id, A_j, Id'>$ et d'après C_3 la table R_i contient un élément $<A_j, Id', n, X, Y>$ avec $Id \in (X,Y)$. Supposons (sans perdre de généralité) que Id = X. Deux cas se présentent selon la valeur de n.

1er cas : n = 2

L'élément $<Id, A_j, Id'>$ est dans ce cas retiré de la table C_i et dans R_i l'élément $<A_j, Id', 2, Id, Y>$ est remplacé par $<A_j, Id', 1, x, Y>$. Les conditions C_1 à C_3 restent vérifiées.

2ème cas : n = 1

Y est alors nécessairement un nombre. L'usager A_i calcule la valeur z = x + Y et envoie la réponse $<A_j, Id', z, \{A_j\}, A_j>$. Les éléments $<Id, A_j, Id'>$ et $<A_j, Id', 1, Id, Y>$ sont retirés des tables C_i et R_i qui vérifient toujours les conditions C_1 à C_3.

Exemple.

On pourrait démontrer par récurrence sur la taille des expressions arithmétiques, que si $<A_i, Id, t, A, A_j>$ est une question, le nombre x figurant dans la réponse $<A_i, Id, x, \{A_i\}, A_i>$ est bien la valeur de l'expression arithmétique t. Nous nous contenterons d'illustrer ceci par un exemple.

Soit A = {A, B, C, D} et soit à calculer l'expression t = (1+2) + (3 + (4+5)). Pour décrire le processus de calcul au cours du temps, les conditions sont supposées idéales :

- tout destinataire effectif prend en compte la question reçue
- le moyen de communication ne cause aucun délai
- le transfert d'un message et son traitement par un usager durent une unité de temps.

Le tableau suivant permet de décrire l'état du système par le contenu des tables des usagers et par les messages émis. Un usager est dit actif s'il est en train de réagir à un message reçu.

N.B. L'affectation des tâches est arbitraire et aurait pu être réalisée autrement.

BHS LILLE

TEMPS		Usager A	Usager B	Usager C	Usager D
0	table R	actif			
	table C				
	Messages	<A, a, (1+2)+(3+(4+5)), A, B>			
1	table R		<A, a, 2, a, b> actif		
	table C		<a, A, a> <b, A, a>		
	Messages		<B, a, 1+2, A, C> <B, b, 3+(4+5), A, D>		
2	table R		<A, a, 2, a, b>	actif	<B, b, 1, 3, a> actif
	table C		<a, A, a> <b, A, a>		<a, B, b>
	Messages			<B, a, 3, {B}, B>	<D, a, 4+5, A, C>

TEMPS		Usager A	Usager B	Usager C	Usager D
3	table R		<A, a, 1, 3, b>	actif	<B, b, 1, 3, a>
	table C		<b, A, a>		<a, B, b>
	Messages		actif	<D, a, 9 {D}, D>	
4	table R		<A, a, 1, 3, b>		actif
	table C		<b, A, a>		
	Messages				<B, b, 12, {B}, B>
5	table R		actif		
	table C				
	Messages		A, a, 15, {A}, A		

Généralisation.

Si les expressions à calculer sont formées de nombres, d'opérations d'addition et de multiplication et si les usagers sont des additionneurs et des multiplieurs tous capables de gérer les protocoles, l'exemple précédent se généralise aisément :

les questions auront pour destinataires potentiels soit les additionneurs, soit les multiplieurs. Lorsqu'un usager crée une question, le destinataire effectif du message est un additionneur ou un multiplieur selon l'opération à réaliser en premier (l'expression est de toute façon analysée par l'usager pour déterminer si elle ne se réduit pas à un nombre) ; il en est de même en cas de délégation.

Le cas où les nombres ne figurent pas tels quels dans une expression arithmétique mais sont repérés par des adresses en mémoire sera traité de la même façon en considérant les mémoires comme des usagers sur lesquels sont réalisées les opérations monadiques Lire et Ecrire qui fournissent respectivement comme services le nombre recherché et une information de bonne exécution.

Pour ces extensions, comme pour d'autres réalisant des fonctions plus complexes, le fonctionnement du système sera plus difficile à décrire mais le principe de base reste le même que précédemment.

IIIC6b Conférence tripartite.

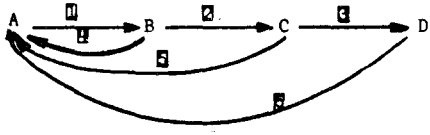
Un usager A (par exemple un processeur) interroge un usager B (par exemple une mémoire) et lui demande de fournir une ressource à un usager C (par exemple un processeur spécialisé) qui utilisera celle-ci. En fin de travail, l'usager C répond à l'usager A. Ce type d'échange se formalise facilement grâce aux délégations généralisées.

$$A \overset{I}{\rightarrow} B \overset{D}{\rightarrow} C \overset{R}{\rightarrow} A$$

IIIC6c La chaîne.

Un usager A interroge un usager B qui d'une part lui répond, d'autre part transmet la requête (généralement c'est la même que celle qu'il a reçue) à un usager C qui d'une part répond à l'usager A, d'autre part transmet à un usager D, etc...

Le mécanisme se schématisera :



Les messages 4, 5 , 6 sont manifestement des réponses adressées à A ; le message 1 est une interrogation, les messages 2 et 3 sont des délégations. Ceci entraîne toutefois qu'une seule requête (contenue dans le message 1) provoque plusieurs services, ce qui est contraire aux définitions posées.

Il faut donc admettre que dans un même message transmis au moyen de communication peuvent cohabiter plusieurs messages logiques, ceux-ci correspondant aux définitions antérieures. Un usager répond alors au message logique qui lui est destiné et regroupe les autres messages logiques dans un même bloc de messages transmis par le moyen de communication.

La représentation logique de cette conférence est un colloque du type :

$$\left\{ \begin{array}{l} A \overset{I}{\rightarrow} B \overset{R}{\rightarrow} A \\ A \overset{I}{\rightarrow} B \overset{D}{\rightarrow} C \overset{R}{\rightarrow} A \\ A \overset{I}{\rightarrow} B \overset{D}{\rightarrow} C \overset{D}{\rightarrow} D \overset{R}{\rightarrow} A \end{array} \right\}$$

De telles structures d'échanges, utilisées par exemple dans une enquête sur l'état des usagers, véhiculent généralement une même requête vers tous les usagers. Le message ⓜ pourra par exemple être formé comme suit :

origine : A, id_1, id_2, id_3
contenu : X
destinataires : B, C, D

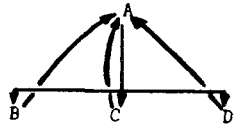
où id_1, id_2, id_3 sont les identificateurs de chacune des consultations. Ceci impose que les usagers soient capables de distinguer les messages logiques formant le message reçu, par exemple en insérant dans celui-ci des séparateurs.

Cette structure d'échanges peut également être formalisée d'autres manières, par exemple en généralisant la notion d'interrogation et en admettant qu'à une seule interrogation peuvent correspondre plusieurs réponses. De telles généralisations sont toutefois hasardeuses car la structuration des échanges, qui est l'un des objectifs recherchés, devient de plus en plus floue ; d'autre part elles entraînent corrélativement une plus grande complexité des mécanismes gérant les protocoles.

Cet exemple simple est plus délicat à résoudre lorsque le nombre d'usagers sollicités par la chaîne est inconnu ; le nombre de réponses est alors défini dynamiquement (i.e. après l'interrogation). Dans ce cas, une interrogation supplémentaire est émise : "quel sera le nombre de réponses qui seront retournées ?" Cette information pourra être connue d'un des usagers sollicités qui répond à l'interrogation correspondante ou ne sera définie que a posteriori. Chaque usager ayant spécifié s'il répond ou non (par exemple grâce à un compteur), le dernier de la chaîne peut répondre à cette interrogation.

IIIC6d L'appel par classes.

Ce type de protocole se retrouve par exemple dans le réseau de Koczela [KOC69] (appel par niveau) ; dans ce cas un usager A envoie un message dans le moyen de communication ; ce message est capté par tous les usagers concernés (i.e. de la classe indiquée) qui le traitent et y répondent.



Il est à remarquer que le moyen de communication permettra (bus par exemple) ou ne permettra pas (processeur cellulaire) ce type de conférence.

Le parallélisme de ce type de conférence le rapproche d'une structure du type colloque :

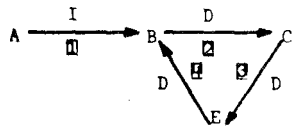
$$\left\{\begin{array}{l} A \overset{I}{\dashrightarrow} B \overset{R}{\dashrightarrow} A \\ A \overset{I}{\dashrightarrow} C \overset{R}{\dashrightarrow} A \\ A \overset{I}{\dashrightarrow} D \overset{R}{\dashrightarrow} A \end{array}\right\}$$

La difficulté est de décrire les interrogations logiques issues de A vers B, C, D en fonction du seul message physique émis, d'autant plus que les destinataires peuvent ne pas être connus de l'usager interrogateur (tant en nombre qu'en identité). La solution semble être de considérer que A envoie des interrogations dont l'identificateur n'est pas totalement défini, les usagers qui répondent complètent chacun de leur côté l'identificateur qui est renvoyé à A. La zone destinataires comprend tous les usagers supposés concernés ; le fonctionnement se formalise alors selon la structure de chaîne, étant entendu que la structure du moyen de communication en simplifie la réalisation.

Par exemple, A envoie le message A, id_o ⌊__⌋, Z, D, X et reçoit de B : A, id_o B, Z, A (où Z est le contenu du message).

III C6e **Bouclage par délégations.**

Exemple : un usager A interroge un usager B ; celui-ci délègue sans travailler à un usager C qui lui-même délègue, toujours sans travailler, vers un usager E qui, dans les mêmes conditions, délègue vers B.



B reçoit donc le message qu'il a déjà manipulé mais dont il n'a gardé aucune trace. Si l'état des usagers B, C, E n'est pas modifié soit de l'extérieur, soit par fin du travail d'un usager, ce bouclage a une durée indéterminée.

Pour résoudre de telles situations, il faut :

- soit qu'un usager qui délègue garde une trace du message manipulé ; ceci encombre les mémoires d'un usager et pose le problème de l'effacement qui devra se faire par une nouvelle conférence (qui elle même pourra boucler) qui parcourra la suite des usagers qui ont délégué (cette suite devra avoir été mémorisée). Ce système paraît inefficace.

- soit conserver les mécanismes décrits en IIIC3 mais adjoindre un mécanisme limitant les délégations où l'usager ne fait pas progresser le travail (délégations "passives"). Ceci pourra se faire par exemple en associant au message un compteur initialisé par tout usager qui a effectivement réduit la partie requête. La valeur initiale représente le nombre maximal de délégations "passives" permises ; ce compteur est décrémenté de 1 à chaque délégation "passive" et dès qu'il est nul, un refus (qui est une réponse particulière) est envoyé à l'usager interrogateur.

N.B. Les délégations où l'usager réduit la partie requête ne peuvent provoquer ce type de bouclage si le contenu du message ne contient pas en lui-même de possibilité de bouclage.

IIIC6f **Bouclage par interrogations**.

Exemple.

Un usager A interroge un usager B ; celui-ci pour satisfaire la requête présentée interroge l'usager A. Dans certains cas, A pourra ré-interroger B par une requête de contenu identique à celui de la requête originelle (mais d'identificateur différent). A ne possède a priori aucun moyen de déceler cette situation qui peut se poursuivre indéfiniment tout en consommant des ressources (temps, moyen de communication, mémoire ...).

La solution de ce type de bouclage peut être l'utilisation d'une hiérarchie (statique ou dynamique) des usagers en classes telles que un usager ne peut interroger que des usagers d'une classe inférieure (par convention) à la sienne ; il pourra même interroger des usagers de sa classe si un ordre y est défini (A peut interroger B dans sa classe si et seulement si le rang de B est inférieur au sien). Il est toutefois probable que cette hiérarchie sera insuffisante pour résoudre tous les conflits (du type "deadlock") et devra être complétée par des mécanismes inspirés de ceux implantés dans les systèmes d'exploitation (sémaphores, moniteurs, etc...) [HAN72], [HOA74].

IIIC7 HIERARCHIE DES USAGERS.

Trois mécanismes de base (Interroger, Déléguer, Répondre) ont été définis précédemment. Un usager peut donc se trouver dans les situations suivantes : il interroge, il est interrogé, il délègue, il répond, il reçoit une réponse (le cas où il reçoit une délégation est identique à celui où il reçoit une interrogation).

Ces différentes possibilités amènent à distinguer des *primitives*. (i.e. des séquences de fonctionnement) liées à la gestion des protocoles.

Nous avons retenu les primitives suivantes :

6 primitives principales

- Interroger (I)
- Recevoir/analyser (A)
- Etre interrogé (E)
- Déléguer (D)
- Répondre (R)
- Recevoir une réponse (S)

2 primitives annexes

- Réveil (W)
- Modification (M)

Le comportement d'un usager vis-à-vis des protocoles est fonction d'une part du répertoire de primitives dont il dispose parmi les primitives citées, d'autre part de la complexité de chacune des primitives qu'il possède. Ceci permet de définir une hiérarchie des usagers.

IIIC7a **Messages**.

Une définition (plus complète que précédemment) d'un message le décompose en quatre parties :

* *origine*
  référence et identificateur

* *informations de gestion*
  ce sont des informations concernant le traitement du message par l'usager, par exemple : le niveau de priorité, le niveau d'interruption, etc...

* *contenu*

* *destinataires*
cette zone comprend d'une part la référence du destinataire effectif du message (i.e. l'usager qui va recevoir le message) d'autre part une spécification de destinataires potentiels. Cette spécification peut prendre différentes formes (liste explicite, spécification implicite , fonction ou application dépendant du contenu du message, de l'usager, etc...) et n'est pas nécessairement exhaustive.

IIIC7b Primitive "Interroger".

Cette primitive qui est activée par le travail exécuté par l'usager doit réaliser les fonctions suivantes :

1) créer l'identificateur
2) stocker l'identificateur et la date de création si elle n'y est pas incluse, gérer les conférences
3) fabriquer la zone origine du message
4) relier l'activité "Interroger" à l'activité qui l'a provoquée
5) fabriquer les informations de gestion et le contenu du message
6) stocker le contenu du message, les informations annexes telles que l'environnement
7) créer la zone "destinataires" et la stocker
8) envoyer le message.

Précisons certaines de ces fonctions :

1) Création de l'identificateur.

Cette fonction est indispensable à tout usager pouvant interroger ; elle pourra être de réalisation très variable selon les usagers puisque la seule contrainte est l'unicité à tout instant du couple référence—identificateur. L'identificateur pourra être plus ou moins riche en information, d'un simple numéro d'ordre à une date de création ou un nom de processus associés à son P.S.W.

2) a) Stockage de la date de création : ceci est utile si un mécanisme de réveil est utilisé par l'usager afin de répéter l'interrogation ou d'activer un mécanisme de contrôle.

b) Gestion des conférences.

Dans la section IIIC5b, il a été vu qu'un usager qui entamme une consultation relie les identificateurs crées à l'identificateur du message reçu antérieurement qui a provoqué la consultation. L'ensemble des réponses reçues avant que l'usager ne reprenne le travail ayant causé la consultation peut ne représenter qu'une partie de l'ensemble des interrogations.

Soit R_n l'ensemble des n interrogations émises par un usager B ; à la réception l'ensemble S_m des m réponses nécessaires peut prendre diverses formes dont :

◊ toutes les réponses sont nécessaires à B

$$\Rightarrow m = n$$

◊ un certain nombre de réponses sont nécessaires, sans distinction individuelle

$$\Rightarrow m < n$$

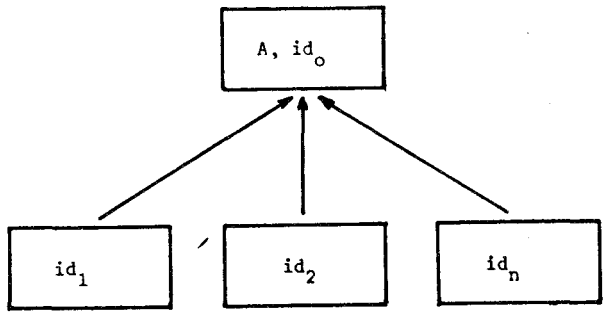
Dès que le seuil de m réponses est atteint, la tâche de B pour A peut se poursuivre

◊ certaines réponses sont nécessaires "individuellement" tandis qu'un certain nombre d'autres le sont sans distinction ; il faudra donc distinguer :

$$\left.\begin{array}{l} R_n = R_i \cup R_b \\ n = i + b \\ S_m = S_j \cup S_c \\ m = j + c \end{array}\right\} \Rightarrow \begin{array}{l} i = j \\ c < b \end{array}$$

S_j : sous-ensemble des réponses indispensables, correspondant à R_i, sous-ensemble des questions ;
S_c : sous-ensemble des réponses non-indispensables, correspondant aux questions R_b.
Il est à noter que la définition de ces sous-ensembles peut être acquise dynamiquement selon les réponses reçues (cf. IIIC7h).

Les combinaisons peuvent être représentées par un graphe orienté où à chaque noeud est associée l'opération à effectuer sur les arcs incidents. Par exemple :



Elles peuvent être gérées simplement en utilisant des graphes bilogiques [BAE] dans lesquels on impose à chaque noeud d'être homogène, i.e. de ne contenir que des opérations logiques de même type (ET, OU). Ceci nécessite toutefois la création de noeuds supplémentaires, fictifs (i.e. ne correspondant à aucune consultation avec d'autres usagers) afin de respecter le formalisme. A ce type de noeud est associé un type particulier d'identificateur.

La mise en oeuvre de cette méthode peut se faire en associant à chaque noeud (et donc à chaque identificateur) un compteur représentant le nombre de réponses à recevoir avant de reprendre le travail correspondant. Sur un noeud ET, ce compteur est initialisé par le nombre de réponses nécessaires, sur un noeud OU il est initialisé à 1. Lors de la réception d'une réponse satisfaisante (i.e. hormis les refus), ce compteur est décrémenté de 1 ; si la nouvelle valeur est zéro, le travail correspondant à id_o peut reprendre.

Exemple.

Soient 5 usagers A, B, C, D, E ; A interroge les autres afin de faire progresser le travail correspondant à (X, id_o)

1) A doit recevoir toutes les réponses. Les relations après interrogations sont représentées sur la figure 1

2) A n'a besoin que 2 réponses : figure 2

3) A a besoin de la réponse de B, de celle de C, et d'une des réponses de D ou E : figure 3.

Les identificateurs id_1, id_2, id_3, id_4 correspondent respectivement aux usagers B, C, D, E.

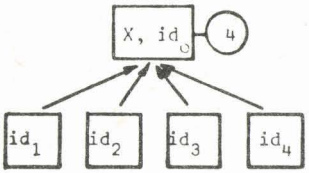


Fig.1 : noeud ET.

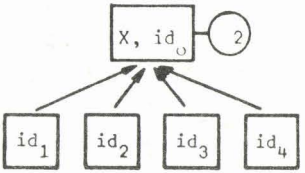


Fig.2 : noeud ET avec seuil

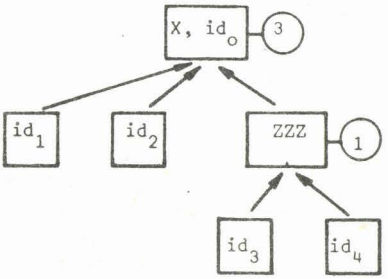


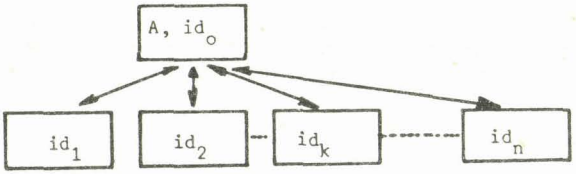
Fig.3 : utilisation d'un noeud fictif (ZZZ) de type OU.

Si la relation de id_i vers id_o est nécessaire, la relation inverse l'est également, par exemple dans le cas suivant. Soit un noeud de type OU ; une réponse portant l'identificateur id_i relié à ce noeud arrive à l'usager. Dès cet instant, les autres identificateurs et informations annexes, les réponses correspondantes aux identificateurs id_j ($j \neq i$) reliés à ce noeud ne sont plus nécessaires au travail représenté par l'identificateur id_o. Deux cas peuvent se produire :

- les réponses correspondant aux id_j sont reçues alors que id_o est toujours connu de l'usager. Leur arrivée entraîne que le compteur associé à id_o devient négatif, moyen simple de détecter leur inutilité. Il n'y a qu'à détruire les identificateurs correspondants et les informations annexes, sauf si ces réponses annoncent une réservation de ressources, auquel cas des consultations de libération de ces ressources sont à débuter

- l'identificateur id_o est détruit alors que des réponses sont encore attendues. Dans ce cas, avant destruction de id_o (et après réception de id_i) il faut avoir marqué les id_j comme désormais sans intérêt et n'entraînant que la destruction des identificateurs et informations annexes (moyennant la réserve vue ci-dessus)

Il faut donc, à partir de id_o, pouvoir accéder aux id_j et le schéma du graphe devient :



La mise en oeuvre des deux relations peut se faire par des tables (cf. IIIC6a) par des listes ou des chaînes, etc...

5) a) Fabrication des informations de gestion.

Elles sont généralement obtenues à partir de l'état antérieur de l'usager ; pour un processeur ce sont par exemple le mode (privilégié ou non), la priorité du processus actif, etc...

b) Fabrication du contenu du message.

Le contenu doit fournir à tous les usagers qui le manipuleront :

1) les données et commandes nécessaires au travail, sous une forme compréhensible par l'usager considéré

2) des éléments nécessaires à la gestion du protocole.

Le premier point implique soit qu'il existe des règles strictes et générales d'élaboration du contenu d'un message, soit qu'un usager qui interroge adapte le contenu du message crée en fonction des usagers qui le manipuleront.

Les diverses possibilités de "compréhension" et d'exécution du contenu d'un message sont précisées pour la primitive "Etre interrogé".

Certains éléments nécessaires à la gestion du protocole seront inclus dans le contenu du message, en particulier des spécifications implicites d'usagers destinataires non répertoriées dans la zone "Destinataires".

7) Création de la zone "Destinataires".

La zone "Destinataires" comprend une spécification des destinataires potentiels qui peut être plus ou moins explicite selon :

◊ les allocations de ressources : si une ressource (usager) est allouée à un usager, c'est à celle-ci (à l'exclusion de toute autre) qu'il devra faire appel. Certaines délégations seront interdites (délégations "passives") ce qui devra être précisé dans la liste des destinataires potentiels

◊ l'équivalence entre plusieurs usagers formant un sous-ensemble de destinataires potentiels qui ont le même comportement vis-à-vis du contenu du message (et donc entre lesquels pourront se faire des délégations "passives").

L'usager qui interroge pourra soit connaître précisément ce sous-ensemble et donc :

- en insérer la liste (éventuellement ordonnée selon un second critère tel que la performance, la charge du moyen de communication, etc...) dans la zone destinataires

/

- préciser les conditions de délégabilité (afin d'éliminer le problème des bouclages signalé précédemment).

soit ne connaître qu'une partie de ce sous-ensemble, auquel cas il devra indiquer dans la zone "destinataires" la partie connue et les possibilités de délégations.

◊ les possibilités des usagers qui manipuleront le message ; s'ils sont à même de déterminer correctement eux-mêmes les destinataires potentiels (en particulier grâce au contenu du message), l'usager qui interroge peut s'en abstenir.

Elle contient également le destinataire effectif ; celui-ci doit être déterminé en fonction du contenu du message afin de respecter l'ordre des opérations à effectuer et les possibilités des différents usagers.

IIIC7c Primitive "Recevoir, Analyser" (A)

Cette primitive, activée par la réception de tout message, a pour fonctions :

1) le contrôle de destination.

Il consiste à vérifier que la référence de l'usager est identique à la référence du destinataire effectif. En cas d'erreur, il faudra d'une part déclencher une procédure de contrôle, d'autre part répondre à l'usager interrogateur par un refus pour cause d'erreur.

2) le contrôle syntaxique.

Cette vérification a pour but de déterminer si le contenu du message est correctement rédigé, si les informations de gestion ne contiennent aucune anomalie flagrante, etc...

3) l'analyse circonstancielle.

En fonction des informations de gestion véhiculées par le message, d'informations sur l'état de l'usager (charge de travail, mode, conférences en cours,...), elle consiste à déterminer si le traitement du message doit être effectué immédiatement, si le traitement est impossible (auquel cas il pourra éventuellement y avoir délégation), si le traitement peut être différé. Dans ce dernier cas, le message est placé dans une file d'attente gérée par l'usager d'où il sera extrait ultérieurement.

4) la détermination du type de message.

Afin de préciser la primitive suivante à activer (S ou E), l'usager doit déterminer s'il s'agit ou non d'un message ne contenant qu'une partie service.

IIIC7d Primitive "Recevoir une réponse" (S)

Activée par la primitive précédente (A), cette primitive a pour fonctions :

1) le contrôle d'erreur : si l'identificateur du message n'est pas connu de l'usager, il y a lieu d'activer une procédure de contrôle d'erreur.

2) la gestion des conférences (cf. IIIC7b).

Le contrôle d'utilité : dans la description de la primitive I (gestion des conférences), il a été vu qu'un identificateur peut être accompagné d'une marque précisant que la réponse est sans intérêt. En ce cas, si le message ne transmet pas une réservation de ressources, la fonction 5 peut être exécutée. S'il y a eu réservation de ressources, l'usager doit :

- mémoriser le message reçu
- débuter des consultations de libération des ressources

inutilement réservées. Ce n'est qu'une fois ces consultations terminées que le message mémorisé pourra être géré.

3) Accès à l'environnement.

Afin de prendre correctement en compte le contenu du message, l'usager accède à la spécification de l'environnement dans lequel il a procédé à l'interrogation ; ceci fait partie des données annexes reliées à l'identificateur.

4) Prise encompte, stockage et éventuellement mise en forme du contenu du message afin de permettre la reprise de l'activité en cours avant l'interrogation.

5) Destruction de l'identificateur et de toutes les données annexes (environnement, destinataires, etc...)

6) Gestion des noeuds fictifs et marquage des inutiles.

Dans la description de la primitive J (gestion des conférences), il a été précisé

a) qu'un compteur associé à chaque noeud (identificateur) spécifie le nombre de réponses nécessaires avant la reprise du travail. Si ce nombre n'est pas nul, le travail n'est pas re-activé (7). Si ce nombre est nul, le travail est à re-activer (7) ; il faut alors marquer, lorsqu'ils existent, les identificateurs correspondant aux consultations dont le résultat est inutile (ceci peut également être effectué par les primitives D et R).

b) que des noeuds fictifs (i.e. ne correspondant à aucun message) peuvent être crées afin de faciliter la gestion des conférences. Si ces noeuds sont satisfaits (compteur nul) et si aucun identificateur ne leur est plus associé, ils peuvent être détruits.

7) Re-activation du travail ayant provoqué la consultation qui vient de s'achever.

IIIC7e Primitive "Etre interrogé" (E)

Activée par la primitive A, la principale fonction réalisée par cette primitive est l'analyse du contenu du message reçu. Tout d'abord, l'usager doit distinguer la partie requête de la partie service. La partie requête est analysée, ce qui conduit à la classification suivante :

◊ requête simple : elle ne contient que des opérations réalisables par un même usager. Elle peut dans ce cas être à un seul niveau (i.e. ne contenir qu'une seule commande) ou à plusieurs niveaux (i.e. contenir plusieurs commandes). Dans ce second cas, les niveaux peuvent être identiques (i.e. concerner la même commande) ou différents (i.e. concerner différentes commandes).

◊ requête complexe : elle contient des commandes à exécuter par plusieurs usagers. Elle est dans ce cas à plusieurs niveaux dont au moins deux sont différents.

Les usagers ne disposent pas tous des mêmes possibilités d'analyse et d'exécution de la partie requête selon la complexité de celle-ci, ce qui amène les points suivants :

1) lorsque l'analyse aboutit à une commande inconnue de l'usager, il doit déclencher un mécanisme de détection d'erreur en renvoyant à l'usager qui a crée l'identificateur un message de refus pour cause d'erreur (tout usager interrogateur devra être capable de gérer ce type de refus).

2) lorsque l'analyse ne découvre aucune commande concernant l'usager, un mécanisme de contrôle d'erreur doit être déclenché.

3) il se peut que l'analyse détecte une requête incomplète, ce qui se produit :

- si des données nécessaires au traitement sont absentes

- si des données sont présentes sous une forme incorrecte et donc nécessitent des opérations préalables. Celles-ci seront soit exécutables par l'usager (on les aura omises parce que leur présence implicite suffit par convention, ou il s'agit d'une erreur), soit exécutables par d'autres usagers

- si des commandes sont absentes vu les données présentes mais ne peuvent être déterminées avec certitude.

Par convention, certaines omissions peuvent être acceptées ; en revanche, d'autres nécessitent soit le déclenchement d'un mécanisme d'erreur, soit l'engagement de consultations afin d'acquérir les éléments absents. Dans ce dernier cas il faut que l'usager soit doté de la primitive Interroger.

4) Selon son état (par exemple le volume de travail en attente) et selon le contenu du message, un usager peut décider soit de participer au travail demandé, soit de ne rien faire. S'il refuse de travailler, la primitive $\mathcal{D}$ est activée (cf. IIIC7f).

Si l'usager accepte de travailler, il stocke l'identificateur du message ainsi que les autres parties afin d'en faciliter la gestion ultérieure ; les éléments du contenu exécutables par l'usager sont extraits et la fin de la primitive $\mathcal{E}$ consiste à activer le travail correspondant.

IIIC7f Primitive "Déléguer" ($\mathcal{D}$)

Cette primitive est appelée soit par la primitive $\mathcal{E}$ dans le cas où l'usager renonce à travailler sur le message (délégation passive), soit par la primitive Répondre. Cette primitive, proche de la primitive Interroger a pour fonction essentielle de construire le message émis par l'usager et donc :

1) d'élaborer la zone origine et la zone des informations de gestion ; ces zones sont obtenues par simple recopie des zones correspondantes dans le message reçu

2) d'élaborer le contenu du message. Ceci consiste à créer un contenu soumis aux mêmes contraintes que pour la primitive Interroger et comprenant :

- les commandes non exécutées par l'usager ; elles sont obtenues à partir du contenu du message reçu ; leur sélection implique une analyse de ce contenu

- les résultats et données obtenus par l'usager qui sont destinés aux usagers qui travailleront ensuite sur le message

- les données non utilisées par l'usager ; celles-ci figurent dans le message reçu et ne sont incluses dans le message transmis que si elles sont nécessaires à la suite du travail.

3) d'élaborer la zone destinataires du message transmis et donc plus précisément :

- de mettre à jour les informations de délégabilité s'il s'agit d'une délégation "passive"

- de supprimer les informations inutiles, par exemple des références de destinataires potentiels qui ne seront plus sollicités car le travail correspondant a été effectué

- d'insérer la spécification des destinataires potentiels utiles qui figure dans le message reçu

- de spécifier si nécessaire de nouveaux destinataires potentiels en fonction du contenu du message et des possibilités de l'usager

- de déterminer le destinataire effectif.

Lorsque aucune possibilité de délégation n'existe, soit à cause de l'allocation des usagers, soit parce que les possibilités ont toutes été épuisées, l'usager génère un ordre de refus et active la primitive R.

4) En plus de ce rôle de création du message transmis, la primitive Déléguer doit :

- marquer les noeuds inutiles si ce n'est pas fait systématiquement lors de la réception d'une réponse (cf.IIIC7 b et d)

- détruire toutes les informations (identificateur, message, etc...) désormais inutiles

- envoyer le message transmis.

IIIC7g Primitive "Répondre" (R)

Cette primitive est appelée soit par la fin du travail de l'usager, soit à partir de la primitive Déléguer.

Dans le premier cas, l'usager accède au message reçu et détermine si la partie requête a été totalement satisfaite ; si ce n'est pas le cas, la primitive Déléguer est activée. Si la partie requête est satisfaite, l'usager doit :

- élaborer les zones "origine" et "informations de gestion" (cf. Déléguer IIIC7f).

- créer le contenu du message en y insérant la partie service obtenue par l'usager.

Dans le second cas, le contenu du message est une information de refus.

Dans les deux cas, l'usager procède ensuite :

1) à la création de la zone "destinataires". Elle ne contient que la référence de l'usager présente dans la zone origine.

2) au marquage des noeuds inutiles si ce n'est pas fait systématiquement lors de la réception d'une réponse

3) à la destruction de toutes les informations inutiles

4) à l'envoi du message.

IIIC7h **Primitives annexes : "Réveil" (W) et Modification (M)**

Primitive "Réveil".

Cette primitive, basée sur l'utilisation de la date de création d'une interrogation, peut être envisagéede diverses façons :

1) afin de réduire le volume de réponses ne présentant guère d'intérêt pour l'usager. Par exemple, lorsqu'un usager (processeur) envoie des informations à stocker par un autre usager(mémoire) la réponse sera soit un avis d'erreur, soit une information de déroulement correct. Si les erreurs sont suffisamment rares, ces réponses peuvent avoir un caractère trop contraignant (charge élevée du moyen de communication par exemple) et une solution plus simple est souhaitée. Celle-ci peut consister à admettre que certains usagers ne répondent que pour signaler des erreurs (lorsque le travail s'est déroulé correctement, la primitive R consiste à détruire l'identificateur et le message stockés) ; réciproquement les usagers qui les interrogent supposent que si aucun avis d'erreur n'a été reçu au bout d'un temps t, aucune réponse ne leur parviendra plus. Ils simulent alors la réception d'une réponse de bon déroulement afin de détruire les informations stockées (identificateurs, environnement, etc...) et de gérer les conférences.

La primitive "Réveil" consiste alors à simuler la primitive S au bout d'un temps t (a la limite, t est nul et donc aucune précaution n'est prise à ce niveau pour se prémunir contre d'éventuelles erreurs)

2) afin de permettre la répétition des interrogations dont les réponses n'ont pas été reçues au bout d'un temps t' après lequel une situation anormale (par exemple une erreur) est présumée. Ce temps t' peut être fonction du contenu du message (i.e. du volume de traitement demandé), des usagers concernés (i.e. de leurs performances), des informations de gestion etc... ; il est calculé lors de l'interrogation (ce serait une fonction supplémentaire de la primitive Interroger).

Après le temps t' (et peut-être après ses multiples entiers si aucune réponse n'est reçue), la primitive W consiste à re-itérer l'interrogation, c'est-à-dire à créer un message soit semblable à celui émis à l'origine (il a été mémorisé), soit en différant par :

- les informations de gestion (par exemple une priorité plus grande peut être automatiquement accordée en cas de répétition)

- la zone "destinataires". L'usager essaie, si c'est possible, de solliciter de nouveaux usagers afin d'augmenter ses chances de recevoir par la suite une réponse.

La répétition d'une interrogation pose des problèmes lors des réceptions des différentes réponses et un mécanisme du type de celui proposé pour la gestion des conférences (cf. IIIC7b) peut être nécessaire.

3) afin d'activer des mécanismes de contrôle d'erreur si la réponse à une interrogation n'a pas été reçue après un temps t".

Cette méthode peut être utilisée seule ou après plusieurs tentatives de répétition selon la méthode précédente.

Primitive "Modification".

Dans la description de la primitive Interroger (cf. IIIC7b), il a été précisé que la structure d'une conférence peut éventuellement être définie dynamiquement, i.e. que selon les réponses reçues il est possible de modifier la définition des ensembles R_i et R_j après l'envoi des interrogations correspondantes. La primitive M qui est appelée par le travail de l'usager (étant donné la complexité et la variété des choix possibles, la définition dynamique ne peut être confiée à des primitives du protocole de communications) a pour rôle de modifier les informations stockées (compteurs associés aux identificateurs par exemple) qui définissent la structure d'une confernece.

IIIC7i Hiérarchie des usagers.

Ainsi qu'il a été précisé au début de cette section, la hiérarchie des usagers est définie statiquement :

1) à partir de leurs répertoires respectifs de primitives

2) à partir de la complexité de chacune des primitives incluses dans leur répertoire

Une définition dynamique doit également tenir compte de l'état de l'usager car celui-ci peut influer sur le répertoire des primitives qu'il peut effectivement utiliser à l'instant considéré.

Répertoires.

Le répertoire d'un usager est essentiellement fonction de sa nature (i.e. des fonctions qu'il remplit dans le système). Ainsi, une mémoire est un composant passif qui ne sera généralement pas doté de la possibilité d'interroger et donc de la possibilité de recevoir une réponse. Les primitives annexes seront également superflues dans ce cas ; selon la complexité de la logique de contrôle associée à la mémoire, celle-ci sera ou non dotée de la possibilité de déléguer.

Même lorsqu'un usager possède les caractéristiques technologiques nécessaires à l'implantation de certaines primitives, il faudra encore que celles-ci soient implantées. Ainsi un processeur pourra ne disposer que d'un répertoire restreint de primitives.

Complexité.

Les descriptions des différentes primitives ont présenté pour chacune une solution maximale, i.e. comprenant toutes les possibilités essentielles. Les primitives mises en oeuvre peuvent ne posséder que certaines des caractéristiques décrites, par exemple dans la primitive *Interroger.*:

- les informations mémorisées et reliées à un identificateur peuvent être plus ou moins nombreuses

- la construction de la zone des destinataires potentiels peut être de complexité variable : liste explicite, algorithme de détermination, etc...

Exemple.

Soit un système comprenant les types suivants d'usagers, caractérisés par leurs répertoires de primitives tant en quantité qu'en complexité :

M1, M2 : mémoires adressables
primitives : *A*, *E*, *R*, *D*
la primitive *D* est supposée ne permettre que des délégations sans modification du contenu du message (délégations "passives")

N1, N2 : mémoires adressables
primitives : *A*, *E*, *R*, *D*
la primitive *D* est ici supposée remplir les deux types de délégation

T : table de correspondance
primitives : A, E, R

P : processeur non spécialisé
primitives : I, S, E, R, A, D

C : opérateur spécialisé dans le traitement de chaînes de caractères
primitives : identiques à celles de N1 et N2.

Les échanges entre les usagers se font en notation polonaise inverse ; une chaîne est de la forme :

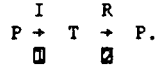
<chaîne> = <{chaîne}*, opération, opérateur> | <valeur>

où "opérateur" est la spécification d'un usager et où "opération" précise la fonction à activer parmi celles dont il est doté. Le nombre de chaînes dans <{chaîne}*, opération, opérateur> est variable selon l'opération demandée (monadique, diadique, ...). Si une seule opération est réalisable par un opérateur, les spécifications "opération, opérateur" sont confondues ; si plusieurs opérations consécutives sont réalisables par un même opérateur, la chaîne s'écrit :

<{chaîne}*, opération 1, ... opération k, opérateur> .

Soit un processus de P qui requiert une opération sur chaînes de caractères portant sur deux chaînes connues par leur nom (qui n'est pas leur adresse) ; la correspondance nom-adresse (dont la référence d'usager) est réalisée par l'usager T. Le résultat est transmis à l'usager P.

La conférence débute donc par la consultation :



Contenu des messages :

7 <P, id_o> : zone origine
<Nom X, Nom Y, Recherche, Recherche, T>: contenu et destinataires

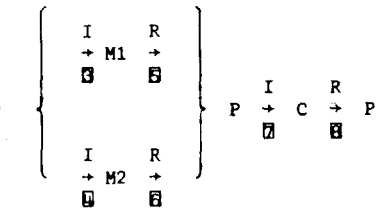
2 <P, id_o>

Divers cas peuvent se présenter, voyons ceux où les mémoires concernées sont N1 et N2 d'une part, M1 et M2 d'autre part:<xN1, yN2> ou <xM1, yM2>

La suite de la conférence a une structure dépendant des mémoires concernées.

a) Les mémoires sont M1 et M2.

La conférence est alors de la forme :



Constitution des chaînes :

3 <P, id1>
<x, Lire, M1>

4 <P, id2>
<y, Lire, M2>

5 <P, id1>
<valeur 1>

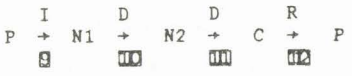
<P, id2>
<valeur 2>

<P, id3>
<valeur 1, valeur 2, opération, >

<P, id3>
<valeur 3>

b) Les *mémoires* sont N1 et N2.

La conférence peut alors être de la structure précédente, mais elle peut également être :



messages échangés :

<P, id4>
<x, Lire, M_1, y, Lire, M2, opération, C>

<P, id4>
<valeur 1, y, Lire, M_2, opération, C>

<P, id4>
<valeur 1, valeur 2, opération, C>

<P, id4>
<valeur 3>

Cet exemple simple montre que la structure des conférences dépend fortement de la hiérarchie des usagers ; il porte toutefois sur des usagers et des opérations simples. On peut généraliser cet exemple à des cas plus complexes où les usagers réalisent des fonctions plus élaborées liées au traitement ou au système d'exploitation (cf. chapitre IV).

IIIC8 CONCLUSION.

Les messages circulant entre usagers selon des structures complexes sont repérés par des identificateurs. Le protocole de communication se ramène pour l'essentiel à des règles simples de création et de gestion de ces identificateurs. Chaque usager effectue lui-même la gestion des identificateurs (et donc des messages) qui le concernent grâce à un petit nombre de primitives définies. Le contrôle du protocole est donc décentralisé ainsi qu'il sied dans une structure multiprocesseur.

IIID MISE EN OEUVRE DES PROTOCOLES.

IIID1 GENERALITES.

Cette section décrit une mise en oeuvre fonctionnelle des protocoles précédemment esquissés (IIIC) sur les niveaux de machines décrits dans la seconde partie de ce chapitre, tant entre niveaux de machines qu'à l'intérieur d'un même niveau.

Les caractéristiques des communications du système multimicroprocesseur envisagé entrent dans le cadre défini en IIIC1 ; les points suivants précisent l'utilisation et la mise en oeuvre des protocoles :

- le moyen de communication est le réseau PROCOPE (cf. ch. II) ;
- la co-opération entre composants du système a été définie au chapitre I ;
- les micro-processeurs et éléments associés (mémoires, unités périphériques) sont les composants du système ; leur puissance limitée, leur diversité nécessitent une grande simplicité des protocoles

◊ si des unités physiques servent d'intermédiaires entre les micro-processeurs et le réseau (et remplissent donc le rôle de certains niveaux de machines décrits), ce sont au plus des objets de possibilités comparables aux micro-processeurs.

Ces caractéristiques ont amené les choix suivants :

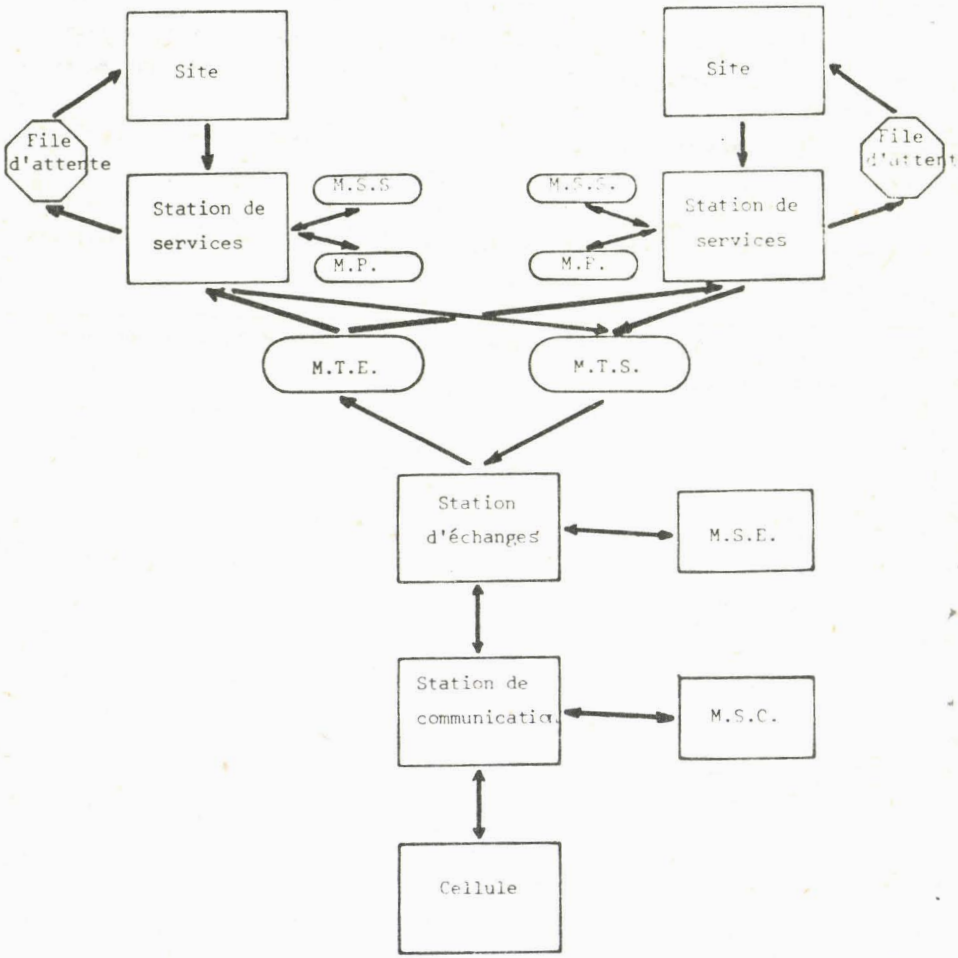
- les transferts sont toujours initialisés par la machine processus, i.e. seuls les composants de la M.P. peuvent interroger. Ils peuvent également répondre et déléguer

- les niveaux intermédiaires (M.E, M.S, M.C) ne peuvent que répondre ou déléguer

- la machine de commutation ne peut que déléguer.

Ceci n'est possible que si le traitement des erreurs détectées aux 4 niveaux inférieurs se fait selon un mécanisme de délégation ainsi qu'il sera précisé plus loin.

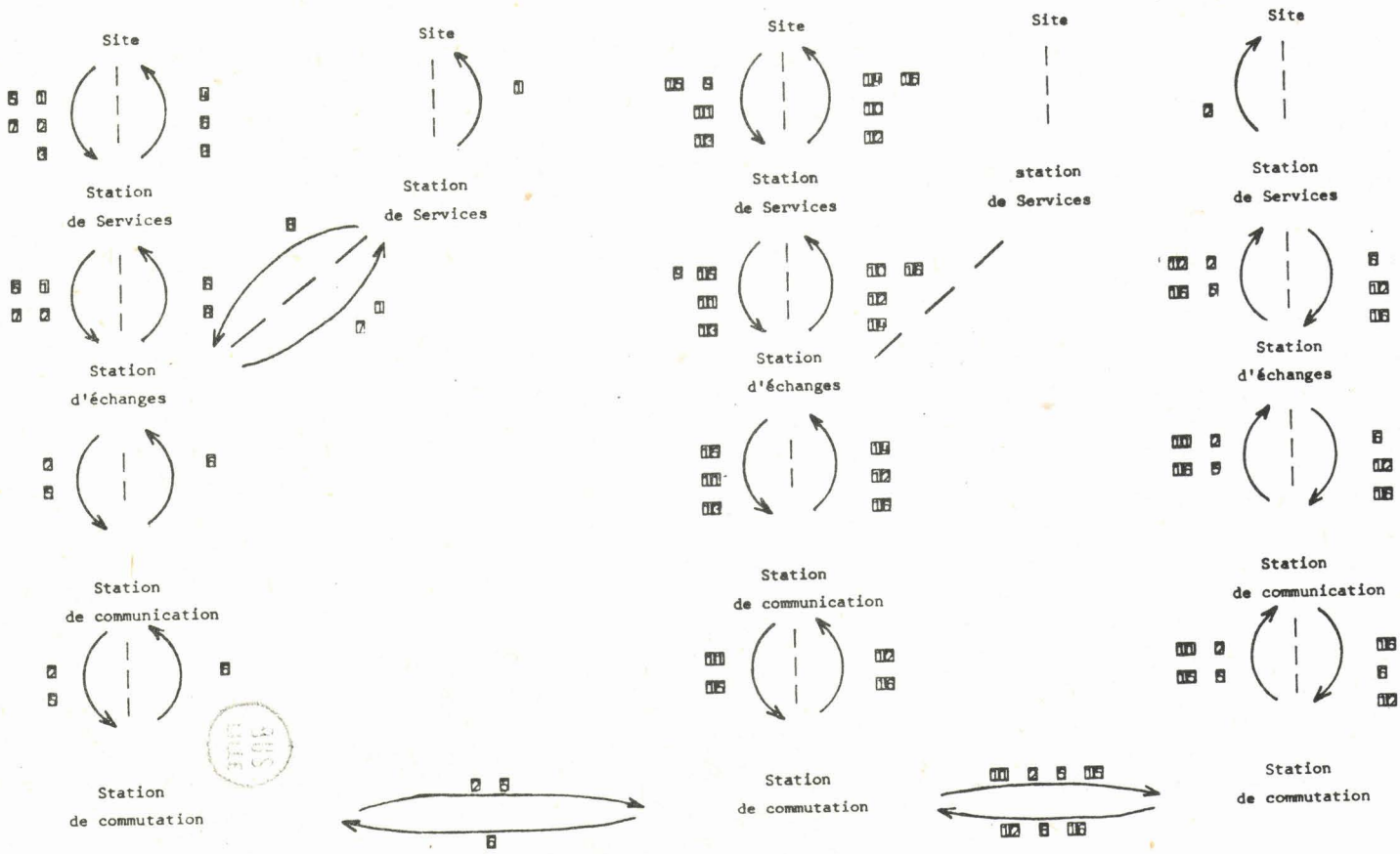
Les schémas suivants résument l'organisation et les différents échanges définis dans cette section.



BPS LILLE

Organisation Fonctionnelle des machines.

TRANSFERTS NORMAUX ENTRE MACHINES



1 Communication entre Sites connectés à une même station d'échanges

2 Communication entre Sites connectés à différentes stations d'échanges

3 Interrogation d'un Site à sa station de services

4 Réponse d'une S.S. au site qui lui est connecté

5 Interrogation d'un Site à une S.S. externe

6 Réponse d'une S.S. à un site externe

7 Interrogation d'un site à une S.S. interne mais autre que la sienne

8 Réponse d'une S.S. à un site interne autre que le sien

9 Interrogation d'un site à sa S.E.

10 Réponse d'une S.E. à un site interne

11 Interrogation d'un site à une S.E. externe

12 Réponse d'une S.E. à un site externe

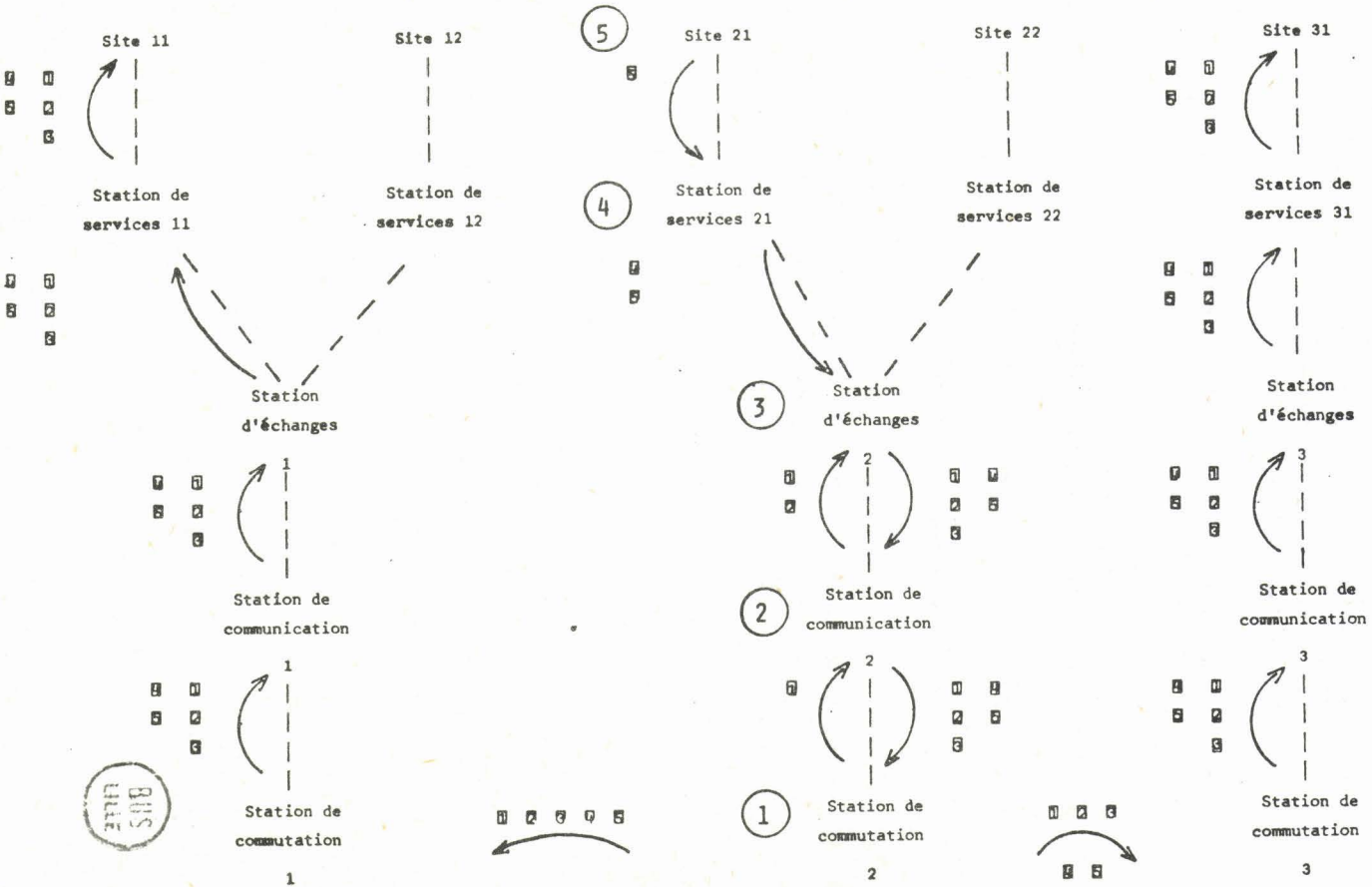
13 Interrogation d'un site à sa S.C.

14 Réponse d'une S.C. à un site interne

15 Interrogation d'un site à un S.C. externe

16 Réponse d'une S.C. à un site externe

GESTION DES ERREURS



1 Erreur de routage (routage impossible)

2 Commande S.C. invalide

3 Erreurs détectées par la station d'échanges :

- commandes S.E. invalides
- contenu erroné
- référence de S.S. inexistante.

4 Erreurs détectées par la station de services

5 Erreurs détectées par le site.

N.B. La mémoire M.S.E associée à la S.E. 2 contient les références des sites 11, et 31 comme étant ceux capables de gérer tous les types d'erreurs.

IIID2 MACHINE DE COMMUTATION (M.R.)

IIID2a Protocole interne de la machine de commutation.

Les protocoles de la machine de commutation sont les règles câblées de fonctionnement décrites dans le chapitre précédent.

Nommage.

L'adressage matriciel du réseau est fixé soit de façon définitive par câblage, soit de façon durable par sélection manuelle, soit par une unité spécialisée pouvant reconfigurer le réseau (par glissement ou par rotation par exemple).

Dans un réseau $n \times n$, l'espace d'adressage est $\{0, n-1\} \times \{0, n-1\}$. Nous nous limiterons aux cas où n est inférieur ou égal à 8, ce qui permet la connexion de 64 unités en l'abscence de multiplexage. L'adressage nécessite donc 6 bits.

Transfert.

Aucune transformation du format des donnés transmises n'est effectuée à l'intérieur du réseau de commutation ; le transfert entre registres s'effectue selon les modalités définies au chapitre précédent ; il peut être considéré comme une forme très élémentaire de délégation. Les seules informations utilisées par une cellule sont

- l'adresse de S.C. destinataire (n° de cellule sur 6 bits)
- le bit de suite associé à chaque élément.

Un mécanisme de contrôle d'erreur est à mettre en oeuvre dans la phase d'analyse d'une locomotive par une cellule afin de détecter les routages impossibles. Ceux-ci peuvent se produire après une erreur de transfert, la perte d'une locomotive, l'éclatement d'un train, etc...(cf. IIF).

Exemple : une locomotive présente dans un registre Nord devrait être transférée dans un registre Ouest.

Pour détecter ces erreurs, une cellule ne dispose que de ses propres coordonnées et de la locomotive qu'elle analyse. Lors de l'analyse, des circuits de comparaison calculent une information concernant le trajet horizontal (H_+ : vers les indices croissants, H_- : vers les indices décroissants, $H_=$: trajet horizontal terminé) et une information concernant le trajet vertical (V_+, $V_=$, V_-). Le contrôle d'erreur consiste à vérifier la cohérence entre ces deux informations et le numéro du registre contenant la locomotive. Les situations suivantes sont seules compatibles :

N et $(H_= \cap (V_= \cup V_+))$

E et $(H_- \cup H_=) \cap (V_+ \cup V_= \cup V_-)$

S et $H_= \cap (V_= \cup V_-)$

O et $(H_+ \cup H_=) \cap (V_+ \cup V_= \cup V_-)$

M et $(((H_+ \cup H_-) \cap (V_+ \cup V_= \cup V_-)) \cup H_= \cap (V_+ \cup V_-))$

Cette procédure garantit la détection de toute erreur impliquant un routage impossible. Une fois l'erreur détectée, la meilleure solution est de forcer la locomotive (et par la suite le reste du train) à sortir du réseau afin de ne pas l'encombrer, voire le bloquer. Un bit de forçage doit donc être associé à chacun des registres N, E, S, O, M ; son état "1" impose un transfert vers le registre R de la cellule (en respectant les règles de fonctionnement déjà définies) ; ce bit reste valide durant le passage du train, sa remise à zéro étant faite après passage du wagon de queue.

En fonctionnement normal comme en cas de panne, aucun mécanisme de contrôle de flux n'est à mettre en oeuvre au niveau du réseau en dehors des règles de fonctionnement précédemment décrites. Ceci est justifié par les points suivants :

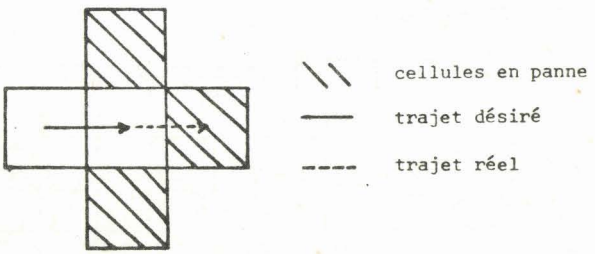
- une cellule doit être un composant passif, simple, rapide (en notation P.M.S. [BEL], ce ne doit être qu'un "switch").

- si la saturation ou la panne d'une cellule devaient être gérées au niveau du réseau, il s'ensuivrait :

que l'état d'une cellule doit être stocké dans les quatre cellules voisines, d'où la présence de moyens de stockage et de mécanismes de mise à jour de ces informations.

que lors du transfert d'une locomotive il faut tester l'état de la cellule destinataire, d'où un ralentissement.

que lorsque l'état est mauvais (saturation persistante, panne) un autre transfert est à rechercher (de préférence dans une direction perpendiculaire). Ce transfert pourra être impossible (cf. figure) ou en contradiction avec la règle de contrôle d'erreur



Pour toutes ces raisons, il nous a semblé que le remède était pire que le mal et donc que la saturation ou la panne d'une cellule (ou d'un registre, ou d'un arc) ne devaient pas être gérées au niveau du réseau de commutation.

IIID2b Protocole externe de la machine de commutation.

1) Machine de commutation - Machine processus (M. R. - M. P.)

* Commande.

Les seules commandes envoyées par la machine processus vers le réseau sont des ordres de reconfiguration par glissement ou par rotation (si ces procédures sont mises en oeuvre).

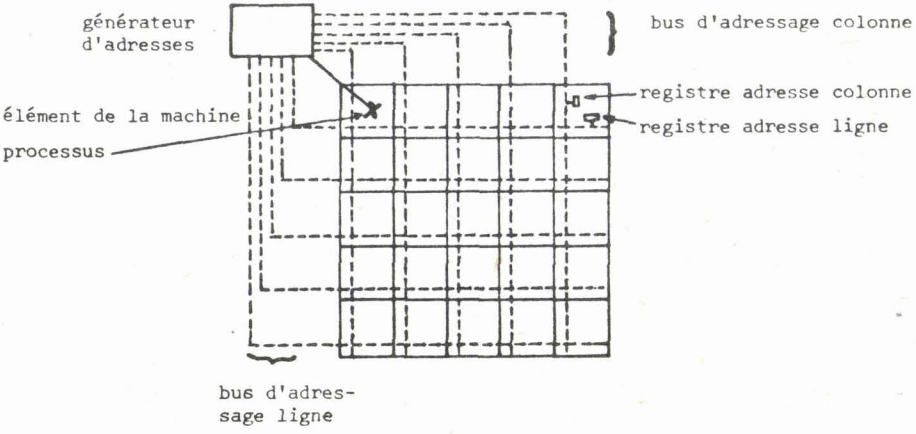
a) *Reconfiguration par glissement.*

Lorsque la machine processus a pu détecter et localiser une panne du réseau, une solution intéressante est de "repousser" la cellule fautive vers la périphérie du réseau, là ou la panne aura une influence réduite voire nulle (par exemple le registre Nord des cellules (1, J) n'est jamais sollicité).

Pour mettre en oeuvre ce type de reconfiguration il faut :

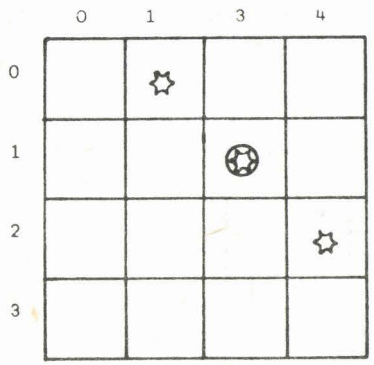
- que le contenu des registres d'adresse des n^2 cellules soit modifiable.

- qu'un organe spécialisé génère les adresses de ligne et de colonne à entrer dans les cellules. Cet organe qui introduit un point de plus grande fragilité est sous le contrôle d'une des unités formant la machine processus.

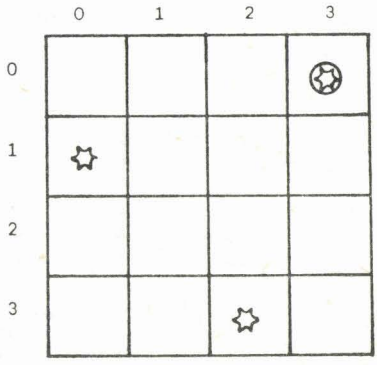


- que le réseau soit physiquement fermé (c'est-à-dire que les connexions entre les cellules extrêmes d'une rangée soient établies).

Dans ces conditions le réseau logique utilisé est un réseau ouvert, d'espace d'adressage (0 - n-1, 0 - n-1) ; voici un exemple de reconfiguration sur un réseau 4×4.



Avant reconfiguration



Après reconfiguration

Après glissement en direction Nord-Est de 1 indice par direction, la cellule d'adresse (1, 2) passe à l'adresse (0, 3) tout en conservant les mêmes connexions avec les cellules voisines (réseau physique fermé).

Remarques.

i) Cette reconfiguration ne permet de gérer qu'un nombre limité de pannes et est insuffisante pour assurer une bonne sécurité de fonctionnement

ii) La reconfiguration ne peut être effectuée qu'une fois le réseau vide de tout message sinon le routage serait perturbé.

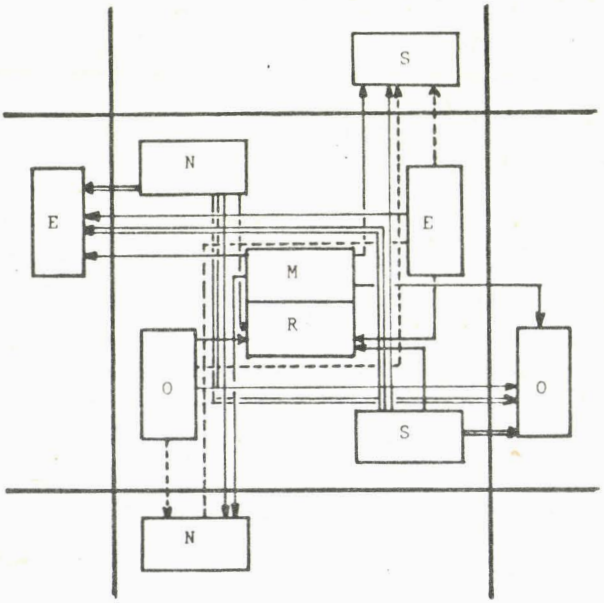
b) Reconfiguration par rotation.

Elle consiste à permuter le rôle des registres "verticaux" (Nord et Sud) et des registres "horizontaux" (Est et Ouest) grâce à une rotation de $\Pi/2$. Elle nécessite :

- qu'un organe spécialisé contrôlé par la machine processus envoie l'ordre de rotation à toutes les cellules grâce à un bus.

- que toutes les cellules puissent prendre en compte un ordre de rotation.

- que les connexions physiquement réalisées entre registres permettent la reconfiguration.



arcs toujours utilisés

arcs définis au chapitre précédent

avec rotation de $\Pi/2$, ces arcs remplacent les arcs en pointillés.

Cette reconfiguration présente les mêmes inconvénients (insuffisance, nécessité de purger le réseau) que la reconfiguration par glissement .

*Livraison.

La machine de commutation ne livre pas directement les éléments à la machine processus. En revanche, elle peut lui livrer des inforamtions sur son fonctionnement.

Pour une cellule, les renseignements disponibles sont :

- l'état d'occupation des différents registres
- l'état de liaison des différents registres
- l'activité de la logique de la cellule: { analyse / transfert } en cours.

Ils peuvent être collectés directement sur les cellules par une unité spécialisée placée sous le contrôle de la machine processus afin d'être utilisés :

BIS LILLE

- pour visualisation sur console
- pour un relevé statistique
- pour la détection du blocage du réseau.

*Contrôle de flux externe ; contrôle d'erreur externe.

La référence prise par la machine processus est l'activité du réseau, fonction du nombre d'analyses et de transferts effectués ainsi que de l'état des registres (occupation, liaison).

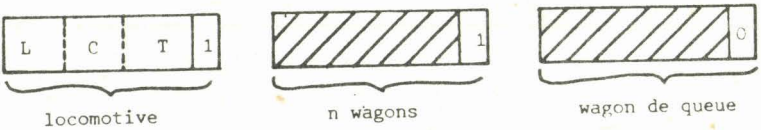
Elle sert à détecter (grâce à l'observation de la M.R.) d'éventuels blocages (en quelques dizaines de cycles le réseau peut être entièrement paralysé) en comparant l'activité du réseau (ou d'une partie du réseau) observée sur une période déterminée (un ou plusieurs cycles) à une activité minimale qui doit se dérouler étant donné l'état des registres.

Lorsque l'activité observée est inférieure à la valeur témoin, la machine processus doit intervenir, soit afin de faire préciser la zone fautive par l'unité d'observation, soit afin d'activer un mécanisme de test après avoir suspendu les injections dans le réseau (cf. IIID4c)

2) Machine de commutation - machine de communication (M.R - M.C.)

* Commande.

Une cellule reçoit un à un les éléments d'un train :



L : numéro de ligne
C : numéro de colonne
} de la S.C. destinataire
T : type de train.

T : cette information précise à quel niveau de machine destinataire est dédié le train ; elle spécifie donc M.P., M.S., M.E., M.C. (cf. IIID5b).

Lorsque le registre M d'une cellule est libre, si la S.C. correspondante a un élément à injecter, elle l'insère dans ce registre et en change le bit d'occupation. Les transferts entre M.R. et M.C. sont des formes élémentaires de délégation (il n'y a même pas de gestion d'identificateur).

* Livraison.

Lorsque le registre R d'une cellule est occupé, il faut :

- soit que l'état 1 du bit d'occupation génère un signal d'interruption vers la M.C.

- soit que la station de communication teste périodiquement (le plus souvent possible afin d'éviter les blocages) le bit d'occupation du registre R.

Une fois l'occupation prise en compte par la M.C., la station de communication { prélève l'élément ; positionne le bit d'occupation à zéro }

* Contrôle de flux.

Intégré à la commande et à la livraison.

* Contrôle d'erreur.

L'erreur à détecter est le routage impossible ; ceci est possible soit si la station de communication teste le bit de forçage associé au registre R indiquant cette erreur, soit si la station de communication re-vérifie l'égalité adresse de S.C. = adresse destinataire contenue dans la locomotive.

IIID3 MACHINE DE COMMUNICATION (M.C.)

IIID3a Protocole interne de la machine de communication.

* Nommage.

Les stations de communication (S.C.) sont en correspondance bijective avec les stations de commutation (cellules). L'espace des noms de S.C. est donc l'espace des noms de cellules ; si des procédures de reconfiguration modifient les noms des cellules, les noms de S.C. doivent être modifiés de la même façon, soit par lecture directe du nom dans les registres d'adresse des cellules, soit grâce à des commandes spécifiques issues de la machine processus.

* Transfert.

Transformations descendante et ascendante du format des données.

Le format d'un message est identique à celui d'un élément, aucune transformation de format n'est donc à mettre en oeuvre à ce niveau si ce n'est pour résoudre les problèmes de routage.

Transfert.

La station d'échanges fournit à la S.C. un message locomotive indiquant (cf. IIID3b) le numéro de S.C. du destinataire effectif (i.e. du site destinataire en général) du train. Les procédures de reconfiguration définies précédemment sont insuffisantes et ne sont pas dynamiques (purge du réseau) ; afin de pallier les pannes du réseau, une procédure de routage adaptatif est donc mise en place au niveau des S.C. . Deux cas peuvent se présenter :

- le destinataire effectif peut être atteint par le réseau, auquel cas l'élément locomotive est identique au message locomotive

- le destinataire effectif ne peut être atteint par le réseau, auquel cas des délégations entre S.C. sont mises en oeuvre pour annuler l'impact de la panne.

Deux catégories de solutions sont envisageables : l'indexation et l'indirection. Dans l'indexation, un train destiné à la S.C. (i,j) transite par la S.C. "relais" (k,ℓ) d'où la S.C. (i,j) est accessible.

Ceci nécessite :

- de marquer le train comme étant à gérer par la S.C. (k,ℓ)
- de fournir les données d'indexation ($\Delta x = i-k$; $\Delta y = j-\ell$) et une commande d'indexation.

Dans l'indirection, le train transitant par la S.C. (k,ℓ) doit contenir :

- la marque "à gérer par la S.C."
- une commande d'indirection.

Il faut également fournir à la S.C. (k,ℓ) la donnée de (i,j) qui est véhiculée par le train.

Ces 2 catégories de solutions peuvent être mises en oeuvre :

- soit par affectation à certaines S.C. d'un rôle exclusif de relais ; la charge de travail correspondante peut alors interdire à la S.C. toute autre fonction, et donc les sites correspondants sont inaccessibles. Cette solution est rapide mais fragile (si le relais n'est plus accessible ?)

- soit sans affectation exclusive du rôle de relais. Toutes les S.C. aptes à jouer ce rôle peuvent être sollicitées, d'où une meilleure adaptation du routage selon la position du site émetteur, une moins grande fragilité, une répartition de la charge de routage, aucune contrainte sur l'utilisation des sites. Cette solution est évidemment préférable.

Plusieurs relais peuvent être nécessaires pour effectuer un parcours donné, par exemple sur la figure 1, la panne de 5 cellules rend impossible l'accès d'un site par une grande partie des autres (cases marquées I) si ce n'est grâce à l'utilisation de deux relais.

Des cas de dégradation importante de l'état du réseau peuvent nécessiter de très nombreux relais (sur la figure 2, pour 20 cellules en panne, il faut une dizaine de relais).

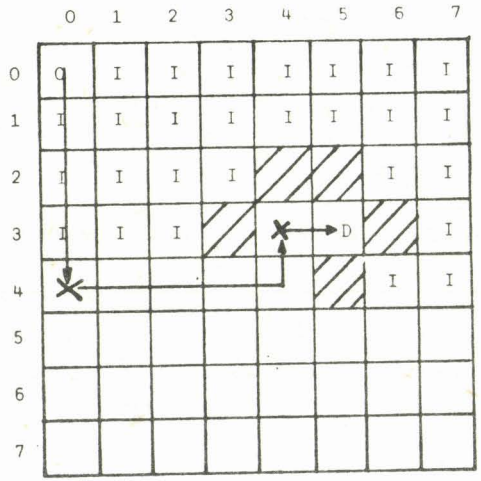


Fig. 1

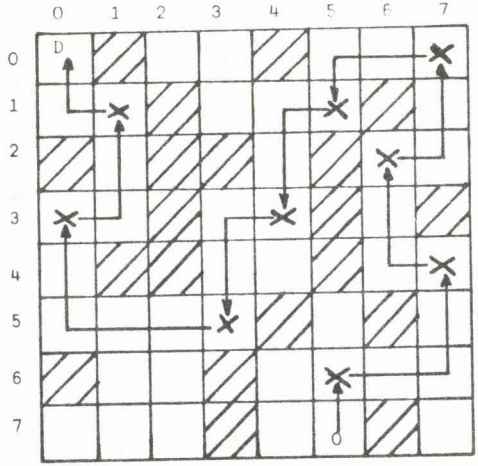


Fig. 2

BNS LILLE

/// cellules en panne

D site destinataire

O site émetteur

S.C. relais

Le parcours du train peut être défini :

- soit entièrement par la S.C. d'origine, ce qui impose que toute S.C. connaisse précisément à tout instant l'état de tout le réseau, qu'elle détermine l'ensemble du trajet, qu'elle crée un train contenant la liste des S.C. relais qui effectueront les délégations

- soit dynamiquement, chaque S.C. relais déterminant quel est le trajet suivant à effectuer en fonction du site destinataire. Dans ce cas, une S.C. ne doit pas connaître l'état de l'ensemble du réseau mais des relations du type "pour aller en (i,j), passer par (k,ℓ)".

La seconde solution est plus intéressante puisqu'elle répartit la charge de routage, qu'elle nécessite moins de stockages et de mises à jour de données sur les routages à effectuer, qu'elle réduit la longueur des trains et donc la charge du réseau (déjà dégradé).

Les possibilités de l'indexation et de l'indirection étant identiques, la seconde méthode est à mettre en oeuvre (elle ne nécessite par de calculs par la S.C.). Pour la réaliser, une mémoire M.S.C. est associée à la S.C. ; elle est accédée par l'adresse figurant dans le message locomotive et contient pour chaque position :

- l'adresse de routage de l'élément locomotive
- une marque de délégation.

Fonctionnement sur la S.C. origine :

1er cas. S'il n'y a pas de marque de délégation, l'adresse de l'élément locomotive = l'adresse du message locomotive = l'adresse stockée dans M.S.C. ; le type de l'élément locomotive = le type du message locomotive.

2ème cas. S'il y a une marque de délégation.

L'adresse de l'élément locomotive = l'adresse stockée en M.S.C.
Le bit de suite de l'élément locomotive = 1. Le type de l'élément locomotive = commande S.C.

Le premier élément wagon = commande de routage.
Le bit de suite du premier élément wagon = 1.
Le second élément wagon = le message locomotive.

Les éléments suivants sont dans les deux cas identiques aux messages reçus de la S.E.

Fonctionnement sur une S.C. relais (i.e. ayant détecté la commande S.C. et une commande de routage) : grâce au second élément wagon, la même procédure que sur la S.C. origine est mise en oeuvre.

Contrôle d'erreur.

Ainsi qu'il a été précisé dans le protocole externe de la M.R., le contrôle effectué par la S.C. est la vérification du routage.

IIID3b Protocole externe de la machine de communication.

* Commande.

La S.C. reçoit de la S.E. à laquelle elle est connectée :

a) des séquences de messages à transmettre par le réseau.

Les messages d'une séquence ont un format identique à celui des éléments d'un train, les bits de suite sont générés par la S.E.

Une séquence comprend :

- un message locomotive précisant la S.C. destinataire et le type de destination (niveau)
- des messages en nombre variable spécifiant :
  + le n° de S.S. destinataire si la M.S. (ou la M.P.) est destinataire
  + les informations de gestion
  + les zones origine, contenu, destinataires potentiels

- un message terminal contenant une information de contrôle, par exemple une "ckecksum" élaborée par la S.E. et portant sur tous les messages transférés entre S.E. (i.e. en dehors de toute considération de routage).

b) des commandes de gestion du routage.

Ces commandes peuvent avoir pour origine soit un site connecté à la S.C. (par l'intermédiaire d'une S.S. et d'une S.E.), soit un site externe (i.e. relié par l'intermédiaire du réseau) car, étant donné les caractéristiques des différents sites, tous ne sont pas capables de générer des commandes. Ces commandes (mise à jour ou consultation de la mémoire M.S.C.) sont transmises par la S.E. quelle que soit leur origine (cf. livraison).

Il s'agit

* soit de mises à jour du type
  "REMPLACER (i,j) par (k,ℓ, x)" qui fait inscrire (k,ℓ) adresse de routage, x marque de routage à l'adresse (i,j) de M.S.C.

* soit de consultations du type
  "ROUTAGE NECESSAIRE POUR ALLER EN (i,j)" dont la réponse est (k,ℓ,x).

* Livraison.

La S.C. livre à la S.E. deux catégories d'informations :

a) des séquences que la S.C. s'est contentée de transférer.

Ces séquences de messages sont destinées à la S.E., la S.S. ou au site.

Ce sera également le cas des commandes de gestion du routage qui sont destinées à la S.C. mais qui sont d'abord transmises à la S.E. afin d'y effectuer un contrôle de validité. La S.E. les décèle grâce à la condition (type de destinataire = M.C.) et (n° S.C. destinataire = n° S.E.) et les transmet à la S.C. qui les exécute puis les re-transmet à la S.E. pour y effectuer la gestion du protocole et la mise en forme nécessaires avant de répondre au site interrogateur.

b) des séquences marquées par la S.C. pour différentes raisons :

- erreur de routage (routage impossible dans le réseau)
- erreur de commande (commande de routage ou de gestion du routage invalide). Dans ce cas comme dans le précédent, la S.C. transmet à la S.E. un message locomotive de type "niveau S.E." suivi d'une commande précisant si possible le type d'erreur puis de la séquence de messages correspondant au train reçu
- commandes exécutées par la S.C. (correspondant au second type de commandes reçues).

* Contrôle d'erreur externe.

Les erreurs externes détectées par la station de communication sont des commandes illégales (mise à jour ou consultation de la mémoire M.S.C.).

Comme les erreurs détectées dans le protocole interne, celles-ci sont transmises à la S.E. (cf. livraison).

IIID4 MACHINE D'ECHANGES.

IIID4a Protocole interne de la machine d'échanges.

* Nommage.

Les stations d'échanges sont en correspondance bijective avec les stations de communication. En cas de modification des noms de S.C. (et donc de S.R.) les noms de S.E. sont donc à mettre à jour.

* Transfert.

La transformation descendante du format des données à la commande consiste :

a) si le destinataire est spécifié par un nom générique (cf. IIID4b), à rechercher dans la mémoire M.S.E. la référence physique du site destinataire et à inclure le n° de S.C. dans la locomotive, le n° de S.S. dans le premier wagon.

b) à opérer si nécessaire une modification de format, i.e. à "découper" le texte reçu en messages acceptables par la S.C.

c) à créer les bits de suite des messages d'une séquence

d) à générer une information de contrôle portant sur l'ensemble de la séquence transmise à la S.C. . Cette information de contrôle constitue le message terminal (bit de suite = 0)

Cette transformation descendante n'est pas à mettre en oeuvre totalement (d) dans le cas de commandes destinées à la S.C. (cf. IIID4b) ou dans le cas de transferts n'empruntant pas les niveaux inférieurs (M.C., M.R.).

La transformation ascendante du format consiste à éliminer les bits de suite, à regrouper si nécessaire les messages en un texte de format acceptable par une station de services, à supprimer l'information de contrôle (si aucune erreur n'a été détectée).

* Contrôle d'erreur interne.

Grâce à l'information de contrôle calculée par une S.E. sur un texte destiné à être transmis par le réseau, une S.E. réceptrice est à même de détecter certains types d'erreur (nouveau calcul de l'information de contrôle et comparaison avec le message terminal reçu). Ces types d'erreurs sont essentiellement ceux où l'information véhiculée par un train a été altérée (découpage d'un train, fusion de trains, altération du contenu d'un élément, etc...).

Lorsqu'une erreur de transmission est ainsi décelée, (ainsi d'ailleurs que lorsque la S.C. transmet à la S.E. des séquences de messages sur lesquelles elle a détecté des erreurs de routage ou de commande), la S.E. les gère directement. Pour cela elle recherche dans la mémoire M.S.E. (cf. IIID4b) la référence physique d'au moins un composant de la M.P. pouvant gérer l'erreur (a priori, l'identificateur et la référence du site interrogateur peuvent être altérés) et elle envoie à la S.C. une séquence de messages comprenant :

- un message locomotive, de type M.P., de référence de destinataire égale au n° de S.C. lu en M.S.E.
- la référence complète du site destinataire (lue en M.S.E)
- une commande d'erreur précisant si possible la nature de celle-ci
- les messages du train erroné

- un message terminal véhiculant une nouvelle information de contrôle calculée par la S.E.
(les bits de suite sont à gérer par la S.E.)

Il est préférable d'alerter deux sites pour gérer les erreurs afin d'augmenter les chances de succès ; ceux-ci sont choisis dans des directions orthogonales de façon qu'au moins un soit en dehors du trajet suivi par le train fautif.

Dans le cas particulier où un site interne à la S.E. est à solliciter pour la gestion d'erreur, un texte est directement transmis à la station service correspondante.

IIID4b Protocole externe de la machine d'échanges (M.E. - M.S.)

* Commande.

Une station d'échanges reçoit des différentes S.S. qui lui sont connectées des textes comprenant :

a) le type (M.P., M.S., M.E., M.C.) du destinataire.

b) la spécification de destinataire effectif.

Après traitement par la station de services, cette spécification peut se faire de deux manières : par une référence physique ou par un nom générique (cf. a) ci-dessous).

c) les zones constitutives (origine, informations de gestion, contenu, destinataires potentiels) décrites dans la section IIIC.

Le destinataire d'un texte reçu par la station d'échanges d'une de ses stations de services peut être :

a) la station d'échanges elle-même.

Ces textes remplissent la condition :
(type de destinataire = S.E.) et (n°S.E. = spécification de destinataire effectif).

Ce sont des commandes de gestion (mise à jour et consultation) de la mémoire M.S.E associée à la station d'échanges. Cette mémoire a pour rôle d'établir la correspondance entre le nom générique (qui est le nom d'une fonction présente dans la machine processus ; par exemple ce sera le nom "unité arithmétique en virgule flottante" ou "composant apte à gérer les erreurs de communication") qui lui est fourni et la référence physique d'un composant de la machine processus.

Dans ce cas, le contenu du texte précise la commande désirée parmi les suivantes :

* "AJOUTER, nom générique, référence physique"
  qui ajoute un couple (nom générique, référence physique) à l'ensemble précédemment stocké

* "DETRUIRE, nom générique"
  qui détruit le couple correspondant au nom générique précisé

* "SUBSTITUER, nom générique, référence physique"
  qui remplace la référence physique associée antérieurement au nom générique par la nouvelle référence

* "LIRE, nom générique"
  qui fournit comme résultat la référence physique associée au nom générique indiqué.
  Si ce nom générique est inconnu, la S.E. doit fournir une information le précisant.

D'après ces commandes, il apparaît intéressant de possèder un accès associatif à la mémoire M.S.E. (accès par nom générique). La mémoire M.S.E. est en particulier mise à jour lors du démarrage du système et en cas de panne soit d'une unité, soit du réseau de commutation (des substitutions peuvent s'avérer nécessaires dans ces conditions). Il faut que la machine processus gère correctement ces mémoires afin de fournir à tous ses composants les références correspondant aux fonctions nécessaires, de garantir une information fiable (en particulier mise à jour), de tenir compte de critères annexes tels que :

- réduire la charge du réseau
- répartir au mieux le travail parmi les différents composants répondant à un même nom générique.

Une fois la commande exécutée, la station d'échanges retourne la réponse à la station de services d'origine. Cette réponse a comme caractéristiques :

- type de destinataire : M.P.
- destinataire effectif : nom de site obtenu grâce à la zone origine
- informations de gestion, zone origine : identiques à celles du texte reçu
- contenu : s'il s'agit d'un ordre de mise à jour, ce sera un écho ou une information de bon déroulement ; s'il s'agit d'une consultation ce sera l'information désirée.

b) une station de services ou un site de même numéro de S.C.

Ces textes satisfont à la condition :
(type de destinataire = M.S | M.P.) et (n° S.C. destinataire = n° S.E.)

Ils ne transitent pas par les niveaux inférieurs (S.C., S.R.) et peuvent être redistribués directement par la S.E. une fois la condition précédente vérifiée (ce qui ne doit être fait qu'une fois la référence du destinataire effectif connu, i.e. après référence à la mémoire M.S.E si nécessaire).

c) la station de communication.

Ces textes satisfont à :
(type de destinataire = M.C.) et (n° S.C. destinataire = n° S.E.)

Ils sont à transmettre à la station de communication sans calcul d'une information de contrôle (mais en exécutant les transformations de format nécessaires).

d) autres.

Dans ce dernier cas, les modalités indiquées dans le protocole interne de la M.E. sont à suivre.

* Livraison.

Les textes transmis par une station d'échanges à une station de services sont :

a) les textes n'ayant pas transité par les niveaux de machines inférieurs (M.C., M.R.) ; ils correspondent aux deux premièrs cas cités dans la commande de la S.E.

b) les textes d'origine externe (i.e. d'une autre S.C.) destinés à un site ou à une station de services.

* Contrôle d'erreur externe.

Les erreurs externes détectées par la S.E. sont :

- des commandes (gestion et consultation de M.S.E.) inacceptables
- les références de stations de services ou de sites qui ne sont pas connectés à la S.E.

Dans ces cas, un mécanisme identique à celui décrit pour le contrôle d'erreur interne est employé.

* Contrôle de flux externe.

Une station d'échanges peut être sollicitée par plusieurs stations de services qui lui sont connectées ; ceci implique :

a) que les textes soumis par les stations de services soient livrés à une vitesse suffisante et donc si possible que la S.S. ne transmette un texte qu'une fois qu'il est complètement élaboré.

b) que la S.E. soit capable d'accepter le plus rapidement possible tout texte qui lui est soumis quelle que soit sa charge de travail. Ainsi elle libère la station de services et indirectement facilite le fonctionnement de la machine processus. Dans ce but, la station d'échanges doit posséder une mémoire M.T.S. (mémoire en tampon de sortie) où la station de services dépose les textes qui sont ensuite traités par la station d'échanges. Un mécanisme du type producteurs-consommateur est mis en oeuvre par les S.S. et la S.E. pour gérer cette mémoire.

Inversement la station d'échanges doit distribuer aux stations de services les textes qu'elle élabore ; une mémoire M.T.E. est placée entre la station d'échanges et les stations de services. Un mécanisme du type producteur-consommateurs est géré par la S.E. et les S.S. afin d'utiliser correctement cette mémoire tampon.

IIID4c Protocole externe de la machine d'échanges (M.E - M.P.)

En cas de panne (par exemple de la machine de commutation), il est parfois utile de suspendre temporairement les communications empruntant le réseau PROCOPE. Cette suspension ne doit concerner que les trains injectés dans le réseau postérieurement à la réception d'un ordre de suspension.

La machine d'échanges étant pourvue de moyens de stockage (M.T.S.) des textes en attente d'expédition, c'est à ce niveau que s'applique ce type d'ordre. Comme cette situation est en particulier rencontrée en cas de blocage du réseau, l'ordre de suspension doit être véhiculé par un autre moyen (par exemple un bus) placé sous le contrôle de la machine processus et dont l'état est accessible par les stations d'échanges.

Lorsque une station d'échanges doit construire une séquence de messages à transmettre par le réseau, elle commence par vérifier qu'aucune inhibition n'est demandée ; si c'est le cas, le mécanisme précédemment décrit s'applique, sinon la S.E. laisse le texte en M.T.S. (sauf s'il s'agit d'un texte ayant trait à un contrôle de l'état du système effectué durant l'inhibition) et se met en attente jusqu'à ce que l'inhibition soit levée.

N.B. Des commandes de suspension et de réactivation peuvent être mises en oeuvre au niveau de chaque S.E. . Leur gestion se fait alors de la même manière que les commandes de gestion de la mémoire M.S.E. .

IIID5 MACHINE DE SERVICES.

IIID5a Protocole interne de la machine de services.

* Nommage.

Plusieurs stations de services sont connectées à une même station d'échanges ; le nom d'une station de services est donc obtenu par concaténation d'un nom de S.E. (de S.C.) et d'un numéro d'ordre repérant la station de services parmi celles qui sont connectées à la station d'échanges.

* Transfert.

La transformation descendante du format des données effectuée par la station de services consiste essentiellement à gérer les protocoles de la machine processus tels qu'ils ont été décrits dans la section IIIC.

C'est sur la station de services que sont implantées les primitives (Interroger, Recevoir- Analyser, Déléguer, Répondre, Etre interrogé, Recevoir une réponse ; Réveil, Modification). Elle gère donc les conférences selon les besoins du site qui lui est connecté et en particulier crée, et gère (mémorisation dans une mémoire M.P., relations) les identificateurs.

Un texte est donc obtenu à partir d'une lettre en effectuant les opérations suivantes :

- création de la zone origine (en cas d'interrogation) ou recherche de la zone stockée (délégation ou réponse) suivie de l'insertion dans le texte.

- recherche de la référence physique du destinataire effectif du texte si celui-ci est connu grâce à un nom logique (cf. IIID5b) la relation nom logique-référence physique sera stockée dans une mémoire M.S.S. associée à la station de services.

- adjonction aux informations de gestion d'informations annexes telles que la nature du texte (ne contenant pas de partie service ou pas de partie requête) afin d'en faciliter la gestion par le site récepteur.

- mise à jour des informations de délégabilité.

- etc...

La transformation ascendante du format des données à la réception consiste à stocker les informations ayant trait au protocole, à les gérer si nécessaire, à prendre en compte les informations de gestion pour déterminer comment sera sollicité le site ...

En particulier si le site n'est pas sollicité immédiatement, il y a création d'une file d'attente entre la S.S. et le site. Cette file d'attente est gérée par une relation de type producteur-consommateur entre la S.S. et le site. Si plusieurs textes logiques peuvent co-exister dans un même texte physique (cf.IIIC6c), la gestion correspondante doit également être effectuée par la station de services.

* Contrôle d'erreur interne.

Ce contrôle d'erreur consiste à vérifier la validité du protocole défini dans la section IIIC. En cas d'erreur, le texte est transmis à la S.E. en étant accompagné d'une commande d'erreur ; le traitement est ensuite du même type que pour les erreurs détectées dans la station d'échanges.

IIID5b Protocole externe de la machine de services.

* Commande.

Les informations constituant une lettre livrée par un site à sa station service sont :

a) le type de destinataire parmi les 4 niveaux de machines accessibles par protocole : M.P., M.S., M.E., M.C.

Cette information n'est nécessaire que si le site interroge, dans les autres cas elle est connue de la station de services.

b) la spécification de destinataire effectif. Cette spécification comprend deux informations.

α) le type de spécification. Il indique l'une des trois possibilités suivantes :

- le destinataire effectif est connu du site par sa référence physique (n° de S.C. ou n° de S.C.|| n° de S.S.). C'est par exemple le cas lorsque cette référence est calculée par le site

- le destinataire effectif est connu du site par une référence logique. Il s'agit dans ce cas d'une allocation et la correspondance référence logique-référence physique est résolue par la station de services

- le destinataire effectif est connu du site par un nom générique. Ce nom générique spécifie une fonction, présente en un ou plusieurs exemplaires, partageable par les composants de la machine processus. Par exemple, un site interroge une unité arithmétique en virgule flottante afin de lui faire exécuter un calcul. Dans ce cas, la correspondance nom générique-référence physique est résolue au niveau de la station d'échanges commune à plusieurs stations de services.

β) la donnée de spécification qui sera soit la référence physique, soit la référence logique utilisée par la station de services, soit le nom générique géré par la station d'échanges.

c) des éléments constitutifs de l'identificateur à créer par la station de services lorsque le site interroge. Il s'agira par exemple d'un nom de processus associé à son P.S.W. ("Process Status Word").

d) des informations de gestion que la station de services insère dans le texte lorsqu'il s'agit d'une interrogation.

e) le contenu

f) des spécifications de destinataires potentiels.

Ces lettres sont soit des commandes destinées à la station de services, soit des informations à transmettre à d'autres composants du système.

Commandes destinées à la S.S. ; elles satisfont à la condition : (type de destinataire = S.S. et référence destinataire = référence S.S.).

Ce sont des commandes de gestion de protocoles (en particulier si la primitive Modification est implantée sur la station de services) ou des commandes ayant trait à la mémoire M.S.S. .

Ces dernières sont :

"AJOUTER, nom logique, référence physique"

"DETRUIRE, nom logique"

"SUBSTITUER, nom logique, référence physique"

"LIRE, nom logique".

Ici encore un accès associatif (par nom logique) serait utile.

Comme pour la S.C. et la S.E., des commandes du même type peuvent avoir pour origine un site différent ; leur transfert à la S.S. est effectué par la S.E. et leur détection se fait selon la même condition.

* Livraison.

La S.S. livre au site qui lui est connecté des réponses aux commandes qu'il lui a adressées et des lettres qu'elle lui transmet.

* Contrôle d'erreur externe.

Il consiste à détecter les commandes inacceptables et à procéder comme en cas d'erreur interne.

IIID6 MACHINE PROCESSUS (M.P.)

IIID6a Protocole interne.

* Nommage.

Les sites sont en correspondance bijective avec les S.S., donc un nom de site est obtenu par concaténation d'un nom de S.C. et d'un numéro de S.S. .

* Transfert.

Des composants de la machine processus sont seuls dotés de la faculté d'interroger ; il existe toutefois une hiérarchie entre sites et seuls certains d'entre eux en sont pourvus. La transformation descendante du format des informations consiste à fournir :

- à la station de services les informations qu'elle a besoin (cf. IIID5b) sous une forme correcte.

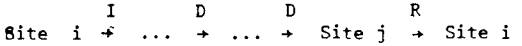
- aux différents niveaux de machines auxquels s'adresse la machine processus des informations exploitables ; en particulier le contenu de la lettre transmise doit répondre aux critères définis en IIIC.

La transformation ascendante du format consiste à remettre les informations sous une forme adaptée aux traitements.

* Contrôle d'erreur interne.

. En cas d'erreur à l'un quelconque des niveaux inférieurs, il y a appel à la machine d'échanges pour déterminer où sera gérée l'erreur. Il en sera de même en cas d'erreur décelée par un site.

Les erreurs aboutissent donc toutes à un ou plusieurs sites (par sécurité). Ce sera ensuite à ce (s) site (s) de prévenir le site ayant généré l'interrogation qu'une erreur a été détectée ; le traitement d'une erreur peut donc être schématisé :



où le site j gère l'erreur ayant trait à l'interrogation du site i.

Cette gestion d'erreur doit bien évidemment tenir compte

a) de la possibilité d'altération de l'identificateur et de la référence du site interrogateur

b) de la possibilité de conflit si plusieurs sites ont été contactés en vue de gérer l'erreur.

Il faudra donc :

a) résoudre le conflit éventuel

b) répondre (par un avis d'erreur) au site interrogateur

c) lancer une procédure de contrôle accompagnée éventuellement d'une suspension temporaire de la création des séquences de messages n'ayant pas trait au contrôle (cf. IIID4c).

IV

ARCHITECTURE ET SYSTÈME D'EXPLOITATION

IVA INTRODUCTION

L'objet de ce quatrième chapitre est d'une part de préciser l'architecture du système multiprocesseur envisagé (section IVB), d'autre part d'ébaucher les aspects principaux d'un système d'exploitation adapté (section IVC), enfin de détailler certains aspects du système d'exploitation (sections IVD et IVE).

Les éléments de choix à prendre en compte résultent du type d'applications envisagées (cf. ID3), du moyen de communications défini (cf. IIE et IIG), des protocoles de communication utilisés (cf. IIIC).

IVB CHOIX D'UNE ARCHITECTURE

IVB1 STRUCTURE DE LA MACHINE PROCESSUS.

IVB1a Microprocesseurs-tâches.

Rappelons que la machine processus définie dans le chapitre précédent est l'ensemble des supports des fonctions de traitement et de gestion du traitement. Il est donc normal de définir sa structure à partir du type d'applications à exécuter, i.e. à partir de la notion de tâche définie dans le chapitre I (cf. ID3g). L'idée de base est d'identifier une tâche à un microprocesseur [ARD75] ; ceci permet une simplification du système d'exploitation puisque l'état d'une tâche est défini par l'état

du microprocesseur qui en a la charge. Etant donné le coût faible d'un microprocesseur et le coût élevé (en temps) d'un changement de contexte, un microprocesseur chargé d'une tâche (noté par la suite M.P.T.) n'est pas préemptible par une autre tâche d'un utilisateur (il peut éventuellement l'être pour une tâche du système d'exploitation telle que la gestion des erreurs de communication).

Les tâches définies au chapitre I amènent le parallélisme à un niveau aussi macroscopique que possible, ce qui permet l'utilisation de primitives de synchronisation évoluées (cf. IVC) avec de bonnes performances. L'interaction entre les microprocesseurs et les autres constituants du système multiprocesseur doit être aussi réduite que possible (la charge de communications et les conflits le ralentissent). Ceci implique d'une part que les références aux mémoires externes à un M.P.T. soient limitées aux données partagées et que la mémoire interne à un M.P.T. contienne une zone de données aussi importante que possible ; en conséquence le langage interne (i.e. les instructions) destiné aux microprocesseurs doit être aussi concis que possible et donc d'un niveau élevé.

D'autre part, pour réduire le volume des interactions le langage de communication entre composants de la machine processus doit lui aussi posséder ces caractéristiques. L'utilisation d'un langage de niveau élevé permet par ailleurs des gains en performance, en sécurité, en temps de programmation grâce à l'utilisation intensive de routine définies et éprouvées.

IVB1b Microprocesseurs fonctionnels (M.P.F.)

Des microprocesseurs réalisant des fonctions relativement complexes semblent nécessaires pour les raisons suivantes :

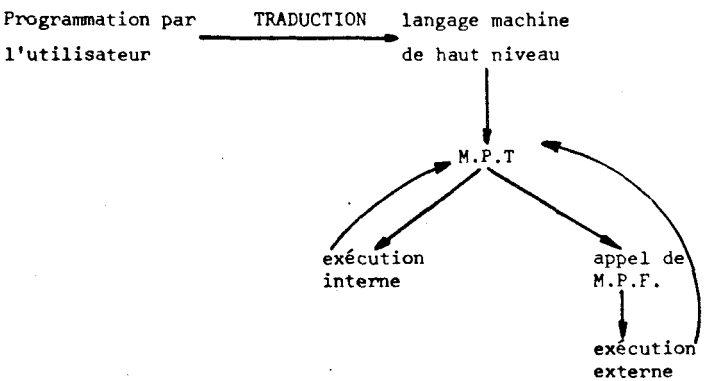
- une tâche ne peut être obtenue par une décomposition trop fine du travail d'un utilisateur (lenteur, coûts de communication et de synchronisation)

- certaines fonctions de traitement évoluées sont souhaitables mais d'une utilisation limitée ; leur mise en oeuvre systématique sur les M.P.T. est difficilement envisageable pour des raisons économiques

- l'extensibilité des fonctions disponibles à l'utilisateur est difficile à réaliser si elle nécessite une refonte de tous les micro-programmes

- certaines fonctions de traitement nécessitent soit une technologie particulière, soit une configuration particulière du micro-processeur.

Cette mise en oeuvre de microprocesseurs fonctionnels (M.P.F.) aboutit au schéma de fonctionnement suivant (proche de la définition donnée dans [DAR74c]).



IVB1c Principales familles de composants.

1) Microprocesseurs-tâches (M.P.T.)

Un microprocesseur-tâche a la charge d'une tâche qu'il exécute en monoprogrammation. Il réalise d'une part le suivi de séquence de la tâche, d'autre part l'exécution des commandes internes et l'appel de microprocesseurs fonctionnels et autres ressources si nécessaire. La détection de ces appels ne peut se faire par interprétation étant donné la lenteur qui s'ensuivrait ; elle est plutôt réalisée par utilisation d'instructions spéciales disponibles sur certains micro-processeurs (par exemple la micro-instruction RST du 8080 [INT76].). A un M.P.T. est associée une mémoire de données et de "programme" (cf. IVB1a). Un M.P.T. a également la charge des fonctions de synchronisation et d'interlocuteur vis-à-vis du système d'exploitation.

2) Microprocesseurs fonctionnels (M.P.F.)

Comme indiqué en IVB1b, des fonctions d'une certaine complexité y sont implantées, par exemple :

- des fonctions d'accès à la base de données
- des opérations arithmétiques en précision multiple ou en virgule flottante
- des opérations de transcodage
- des fonctions de gestion d'environnement (protection...)
- etc...

N.B. Il s'agit soit de routines stockées en "R. O. M.", soit en "R. A. M." ce qui permet de les recharger en cas de besoin.

Certaines fonctions partagées par les tâches telles qu'une horloge en temps réel sont également implantées sous forme d'unités fonctionnelles (qui ne sont pas nécessairement toutes des micro-processeurs).

Les protocoles décrits en IIIC sont évidemment des moyens d'accès adaptés à l'utilisation des unités fonctionnelles.

3) Mémoires.

Afin d'une part de stocker les données communes à plusieurs tâches (données partageables ou transmises), d'autre part de réaliser un niveau intermédiaire de hiérarchie entre les mémoires de masse et les M.P.T., des mémoires externes aux M.P.T. sont à mettre en oeuvre ; des possibilités de transfert par blocs permettent de réduire la fréquence des références à celles-ci.

4) Microprocesseurs d'entrées-sorties.

Ces unités chargées d'une part des opérations d'édition et de lecture sur périphériques, d'autre part du stockage en mémoire de masse ont des caractéristiques voisines de celles des M.P.F. . Des commandes évoluées d'édition y sont par exemple implantées.

IVB2 REALISATION DES NIVEAUX DE MACHINES.

IVB2a Machine processus.

Les composants de la machine processus sont des microprocesseurs et circuits annexes faisant partie des mêmes familles de composants. Citons les microprocesseurs usuels suivants : 8080 d'INTEL, 6800 de Motorola, 2650 de Signetics, F8 de Fairchild qui possèdent des répertoires évolués d'instructions. Le F8 paraît particulièrement bien adapté à la constitution d'unités fonctionnelles étant donné sa mémoire "scratchpad" de 64 octets.

IVB2b Autres niveaux de machines.

* Caractéristiques nécessaires.

Pour réaliser les autres niveaux de machines (M.S., M.E., M.C.) décrits au chapitre III, les caractéristiques suivantes sont nécessaires :

- accès à l'état, à l'adresse, au contenu des registres émetteur et récepteur de la cellule (S.C.)

- mémoires (M.S.S., M.S.E., M.S.C.) de gestion des communications décrites dans la section IIID.

- interprétation de commandes

- opérations logiques (comparaisons) et de calcul (e.g. "checksum" calculée par une S.E.)

- mémoires de travail et de stockage temporaire. Il faut noter que les composants de la machine processus ont une puissance limitée, que les composants fonctionnent en monoprogrammation et donc se mettent souvent en attente après une interrogation. Ces mémoires sont donc de capacité limitée.

De ces caractéristiques il résulte que, globalement, ces niveaux de machines ont des besoins voisins de ceux d'un processeur.

* Configurations.

Diverses configurations peuvent être envisagées selon les besoins en communications (débit, vitesse) des composants de la machine processus. Elles sont schématisées ci-après.

Lorsqu'un seul site est connecté à une station de communication, trois cas sont à envisager.

a) Le site est un micro-processeur pouvant gérer les communications (fig. 1).

b) Le site n'est pas un micro-processeur (c'est par exemple une mémoire), un micro-processeur est donc placé en interface entre le site et le réseau PROCOPE (fig. 2).

Ce micro-processeur doit être rapide (afin d'une part de ne pas ralentir le site, d'autre part d'éponger le contenu du registre récepteur à une cadence suffisante) mais il n'a pas besoin de posséder un jeu d'instructions étendu. Selon les cas un micro-processeur évoqué en IVB2a sera utilisé ou une unité plus rapide (INTEL 3000, 2900 AMS, 10800 Motorola) sera mise en oeuvre.

c) Le site est un micro-processeur mais il ne peut gérer les communications, soit parce que sa charge de travail ne le permet pas, soit parce qu'il ne peut être interrompu, soit parce que sa technologie ne le permet pas (par exemple s'il est trop lent). Généralement il gère toutefois les protocoles et l'allocation, d'où le schéma de la figure 3.

Lorsque plusieurs sites sont connectés à une même station d'échanges, deux grandes catégories sont à distinguer.

1) Aucun des sites ne peut gérer les communications aux deux niveaux inférieurs. Certains peuvent toutefois gérer leurs protocoles, ce qui aboutit à la figure 4.

2) Un des sites gère les communications aux deux niveaux inférieurs ; il peut gérer les protocoles (S.S.) de certains sites qui n'en ont pas la capacité. (fig. 5).

Le choix des sites connectés à une même station d'échanges doit être effectué en tenant compte de la charge admissible par le composant jouant le rôle d'interface afin de ne pas le saturer.

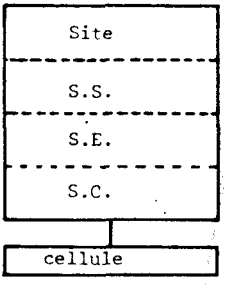


Fig. 1

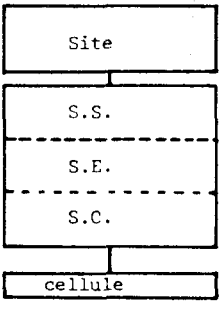


Fig. 2

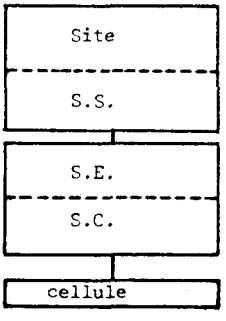


Fig. 3

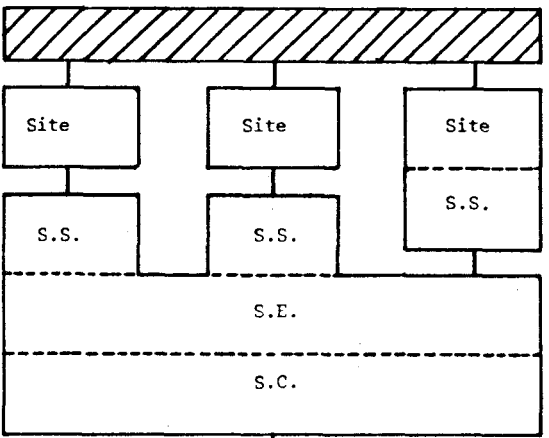


Fig. 4. cellule (station de commutation)

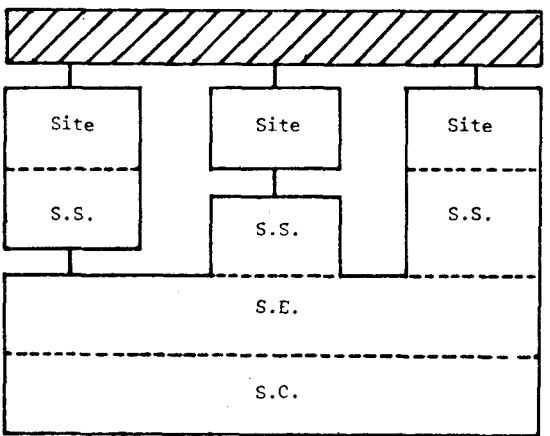


Fig. 5. cellule (station de commutation)

N.B. Les zones hachurées représentent d'éventuelles ressources communes à plusieurs sites, e.g. horloge ou R.O.M. communes à plusieurs micro-processeurs.

BUS LILLE

IVC STRUCTURE GÉNÉRALE DU SYSTÈME D'EXPLOITATION.

Dans la conception d'un système informatique, l'idéal serait de concevoir l'architecture et le système d'exploitation dans une même démarche afin d'obtenir de l'ensemble des performances proches de l'optimum.

Cette section vise à proposer des mécanismes généraux permettant de faire fonctionner cette architecture dans les conditions décrites au chapitre I tout en limitant la charge que représente le système d'exploitation.

IVC1 CONTRAINTES GENERALES.

Deux grandes directions [ENS75] s'offrent dans la conception du système d'exploitation d'un multiprocesseur : la centralisation et la décentralisation. La centralisation présente de graves inconvénients, en particulier la fragilité aux pannes, la lenteur par présence d'un goulot d'étranglement. La décentralisation peut être envisagée soit par dissémination (un système d'exploitation par processeur) soit par traitement symétrique [ENS75] où les fonctions du système d'exploitation "flottent" d'un processeur à un autre selon les besoins. Cette dernière solution est la plus délicate à mettre en oeuvre mais c'est également la plus prometteuse tant en rapidité, qu'en fiabilité... Les principales difficultés rencontrées concernent les conflits, la synchronisation, le partage de ressources (d'une part des besoins peuvent être simultanés, d'autre part certaines fonctions du système d'exploitation ne peuvent exister en plusieurs exemplaires à un même instant).

Ce type de solution a été choisi et donc le système d'exploitation n'a pas d'affectation fixe (à l'inverse par exemple de [ARD75] où des processeurs de service ont pour rôle l'exécution du système d'exploitation). Un processeur tente d'exécuter une fonction système lorqu'il en en a besoin ; des règles d'exclusion mutuelle sont toutefois à respecter et l'exécution effective d'une fonction système ne sera permise qu'après leur satisfaction.

IVC2 CONTRAINTES PARTICULIERES.

Rappelons quelques contraintes particulières au système envisagé. Elles concernent d'une part le réseau de communications, d'autre part les processeurs.

IVC2a Le réseau de communications.

La principale difficulté est l'asynchronisme total des unités connectées, ce qui est concrétisé d'une part par le délai de transmission qui est variable et non négligeable, d'autre part par le parallélisme permis. En particulier, rappelons qu'une structure de réseau permet des échanges simultanés d'un objet A vers un objet B et de l'objet B vers l'objet A.

La possibilité d'erreurs et de pannes et la fragilité du réseau (cf. IIF) rendent nécessaires des mécanismes de reconfiguration ; ceux-ci ne doivent pas perturber le système d'exploitation.

IVC2b Les processeurs.

Tout d'abord, le nombre de processeurs connectés est très élevé (sur un réseau 8×8, il y a 64 stations d'échanges dotées chacune de possibilités de multiplexage) ; il est de plusieurs dizaines au moins. Ceci entraîne d'abord que les conflits et interactions entre processeurs ou entre processeurs et autres ressources peuvent réduire considérablement les performances (qui ne croîtront pas linéairement avec le nombre de processeurs connectés). Vu le nombre de processeurs, la fiabilité est recherchée non pas à l'intérieur d'un processeur mais par présence de plusieurs exemplaires équivalents ; le nombre d'unités en état de marche doit donc être considéré comme variable.

Les processeurs sont des micro-processeurs, donc des objets à coût relativement faible. Ce fait associé à leur nombre a une très grande influence sur le système d'exploitation : d'une économie de pénurie, le système tend vers une économie d'opulence. Il y a donc possibilité de gaspiller ces ressources au profit d'une simplification du système d'exploitation. (Cette optique est différente de celle de Flynn dans [FLY] car l'objectif qu'il s'y fixe est de maximiser la performance par dollar investi).

Enfin, ces processeurs nombreux et économiques ont des performances individuelles faibles ; le système d'exploitation doit donc soit être suffisamment simple, soit être segmenté de façon d'une part à ne pas excéder les possibilités d'un micro-processeur, d'autre part à ne pas monopoliser les ressources de la machine. Cette approche exclut des systèmes d'exploitation très sophistiqués tel celui présenté dans [BLE76].

IVC3 SYSTEME D'EXPLOITATION ET MACHINES EPHEMERES.

IVC3a Rappels.

Le but du système d'exploitation (cf. IE) est de constituer à la demande :

- pour une durée limitée au déroùlement de la tâche demandée
- avec la meilleure configuration possible pour cette tâche

une machine complète, capable de mener à bien le travail qui lui est confié.

Rappelons qu'une tâche est caractérisée d'une part par des besoins implicites qui concernent les ressources du système d'exploitation, d'autre part par des besoins explicites qui concernent :

- des ressources matérielles : processeurs, mémoires, ...

- des ressources logicielles : programmes, routines, ...

- des ressources utilisateur : données fournies à une tâche, informations de synchronisation, ...

IVC3b Machines éphémères.

Définition.

Une machine éphémère (M. E.) est une association de ressources constituée afin de mener à bien une tâche d'un utilisateur.

Cette définition générale peut être précisée par les points suivants : Il s'agit d'une association à laquelle est fixé un but ; en plus d'un groupement de moyens (les ressources) cette notion de machine éphémère implique donc des règles de fonctionnement (commandement, observation, etc...).

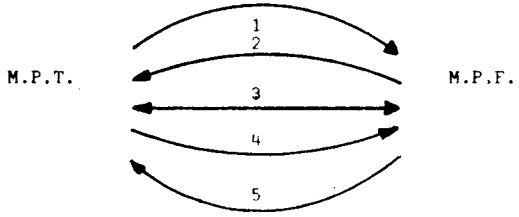
La notion de machine éphémère est identifiée à la notion de tâche, elle est donc reliée à la notion de micro-processeur-tâche. Celui-ci, toujours présent dans une machine éphémère, joue vis-à-vis des autres composants le rôle de noyau de contrôle et de synchronisation ; en particulier il a la charge des relations avec le système d'exploitation et avec d'autres machines éphémères (partage de ressources, conflits, etc...).

Une machine éphémère contient, en plus du microprocesseur-tâche, les autres ressources (mémoires, processeurs d'entrées-sorties, processeurs spécialisés) qui sont reliées à celui-ci par des relations hiérarchiques.

Ces constituants peuvent être :

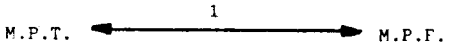
- alloués à la machine éphémère, soit exclusivement, soit avec des mécanismes de partage

- à allocation dynamique en cours de tâche, lorsque le besoin s'en fait sentir. Deux catégories d'allocation dynamique (qui concernent les micro-processeurs fonctionnels) sont envisagéesselon la structure de la conférence : avec ou sans réservation préalable. La première est utilisée lorsque le M.P.F. doit mémoriser des informations entre plusieurs interrogations d'utilisation.



1) interrogation : demande de réservation
2) réponse : réservation effectuée
3) conférence d'utilisation
4) interrogation : libération de ressource
5) réponse : libération effectuée

Allocation dynamique avec réservation.



1) conférence d'utilisation

Allocation dynamique sans réservation.

Comme il a été indiqué dans le premier chapitre, les besoins d'une tâche en ressources matérielles sont précisés sous une forme maximaliste spécifiant en particulier la configuration permettant le parallélisme. Selon les disponibilités en ressources, ces possibilités seront ou non mises en oeuvre. Pour cette raison et parce que la réalisation d'une ressource peut éventuellement être de nature variable (micro-programmée, câblée, etc...), pour une même tâche la machine éphémère peut prendre diverses configurations.

IVC3d Structure générale du système d'exploitation.

Nous avons distingué sept fonctions principales dont six du système d'exploitation :

1) la fonction de gestion des tâches (F.G.)
2) la fonction allocatrice (F.A.)
3) la fonction constructrice (F.
4) la fonction de surveillance (F.S.)
5) la fonction récupératrice (F.R.)
6) la fonction magasinier (F.M.)
7) la fonction utilisatrice (F.U.)

1) La fonction de gestion des tâches.

Elle a pour fonction de gérer les tâches des utilisateurs et donc :

- de prendre en compte les événements fournis par l'extérieur demandant l'exécution d'une transaction.

- de gérer les tâches qui n'ont pas encore été activées, soit parce qu'elles ne sont pas candidates à l'exécution par manque de ressources utilisateur (qui sont la base de l'enchaînement entre tâches), soit parce que bien que candidates à l'exécution, des ressources demandées et indispensables (logicielles ou matérielles) ne sont pas disponibles. Elle doit en particulier déterminer quelles sont les tâches qui deviennent

candidates à l'exécution pour lesquelles elle fournit à la fonction allocatrice une demande et des spécifications du type "APPLIQUER (fonction, objets, espace paramètre)" [DAR74].

- de gérer les tâches dont l'exécution est terminée en prenant le contrôle des ressources utilisateur que fournit la fonction récupératrice afin de déterminer les tâches à activer (et de leur transmettre l'espace paramètre correspondant), les tâches à créer (et à nommer), les tâches à détruire (leur exécution est annulée).

Les micro-processeurs-tâches n'étant pas préemptibles par des tâches, la fonction de gestion des tâches ne doit pas être active à tout instant. Il suffit qu'elle soit mise à jour, par exemple par un mécanisme du type P.I.D. ("pseudo-interrupt-device) [FUL73] avec file d'attente et qu'un micro-processeur en attente de travail consulte ce P.I.D. et exécute si nécessaire la fonction de gestion de tâches.

Cette fonction de gestion des tâches peut être construite à partir de moniteurs et de régions critiques conditionnelles [HAN72], [HOA74], [HANS72],[CAM] ; en particulier dans [SCH76] est proposé un mécanisme simplifié grâce à l'utilisation de relations invariantes entre moniteurs.

2) La fonction allocatrice.

A partir des demandes fournies par la fonction de gestion des tâches (chaque demande comprend d'une part la spécification des ressources demandées, qu'elles soient allouées ou non, et la spécification des ressources fournies telles que l'espace paramètre), la fonction allocatrice doit pour chaque tâche :

- rechercher, grâce à la fonction magasinier, des ressources nécessaires à la tâche et les réserver.

- déterminer si, étant données les ressources recueillies, la tâche peut être exécutée (i.e. si une configuration minimale a été obtenue).

- choisir parmi les ressources recueillies celles qui vont constituer la machine éphémère. Ce choix est effectué en fonction de critères de performance, d'une part des ressources, d'autre part des interactions entre ces ressources ; en particulier le principe de localité évoqué en IIC1 devra être recherché à ce niveau.

- libérer les ressources inutiles

- transmettre les ressources réservées et la demande à la fonction constructrice.

3) La fonction constructrice.

A partir des informations fournies par la fonction allocatrice, cette fonction constructrice doit d'une part créer les liens entre ressources allouées à une machine éphémère (la création de ces liens consiste essentiellement en une mise à jour de la mémoire M.S.S. décrite en IIID5, voire de la mémoire M.S.E.), d'autre part faire vérifier la viabilité de la configuration par la fonction de surveillance (F.S.).

Une fois ces opérations réalisées, la fonction constructrice doit mettre les ressources matérielles dans un état correct, i.e. leur fournir si nécessaire les ressources logicielles (chargement du "programme" par exemple). Enfin, elle active la machine éphémère par un ordre du type "INTERPRETER" (fonction, objets, espace paramètre)" [DAR74].

4) La fonction de surveillance.

Cette fonction de surveillance assume deux rôles :

- rôle préventif. Il s'agit d'effectuer des tests à la demande de la fonction constructrice afin de valider une configuration

- rôle curatif. Il s'agit de détecter les erreurs supputées par les autres fonctions, de prendre les mesures de reconfiguration nécessaires (par exemple en mettant à jour les mémoires M.S.C. et M.S.E. décrites en IIID) et d'en avertir la fonction magasinier, les fonctions récupératrice et de gestion des tâches.

Ces tests concernent aussi bien l'état des ressources matérielles de la machine processus que des machines de niveauxinférieurset en particulier du réseau de commutation (cf. IVD).

5) La fonction récupératrice.

En fin de tâche, il s'agit d'une part de récupérer des ressources libérées (matérielles et logicielles) pour les restituer à la fonction magasinier en leur affectant si nécessaire un état permettant leur utilisation ultérieure. D'autre part des ressources (dont les ressources utilisateur) à transmettre à d'autres tâches sont placées sous le contrôle de la fonction de gestion des tâches.

6) La fonction magasinier.

Comme son nom l'indique, cette fonction tient à jour l'inventaire des ressources disponibles.

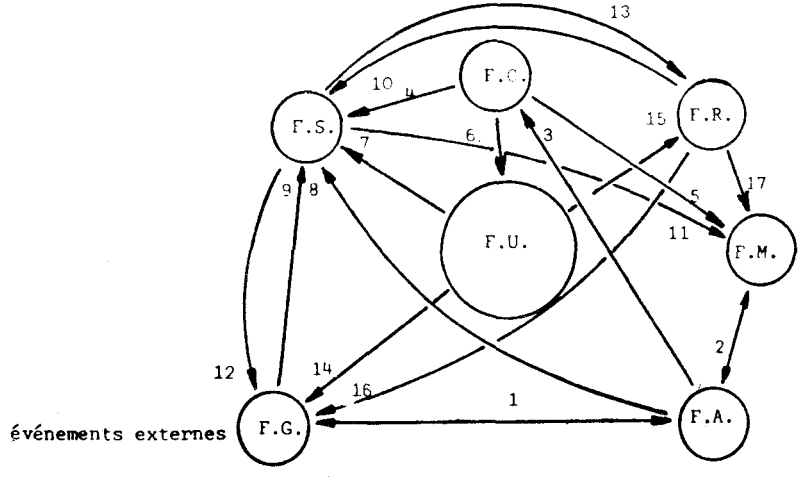
7) La fonction utilisatrice.

C'est l'ensemble des machines éphémères que contient le système à un instant donné, il représente donc les tâches actives.

Les hypothèses (cf. IVC1, IVC2) de réalisation du système d'exploitation impliquent de permettre d'une part que ces fonctions puissent exister simultanément, d'autre part que plusieurs exemplaires de chaque fonction puissent exister simultanément. La cohabitation entre fonctions ne pose pas de grandes difficultés, en revanche à l'intérieur d'une même fonction le problème est plus complexe.

Les fonctions constructrice, récupératrice, de surveillance et de gestion des tâches peuvent exister en plusieurs exemplaires moyennant quelques précautions de synchronisation, de protection.

La fonction magasinier et la fonction allocatrice posent de délicats problèmes de mise à jour (il faut éviter qu'une ressource soit considérée comme libre et occupée par deux magasiniers différents), d'exclusion mutuelle, de conflits (un allocateur pourrait réserver temporairement une ressource qu'il n'a pas besoin et en priver un autre), etc... d'autant plus que les composants fonctionnent en asynchronisme total. Nous avons donc choisi, d'une part de rendre unique la fonction magasinier en la distribuant à travers la machine (chaque composant stocke son état, ou le fait stocker par sa station de services) d'autre part de n'admettre à un instant qu'un seul objet allocateur (ceci supprime les délicats dialogues de négociation lorsque deux objets allocateurs sont en conflit).



1 Transmission des tâches à activer et retour éventuel en cas de ressources indisponibles.

2 Recherche de ressources, choix, réservation.

3 Transmission des ressources et des spécifications de machine éphémère nécessaires à la construction.

4 Contrôle de configuration préalable à la construction.

5 Recherche de ressources logicielles (programmes, routines ...) à charger sur les ressources matérielles.

6 Activation de machine éphémère.

7
8 Erreurs supposées, transmises à la fonction de surveillance
9 pour vérification.
10

11
12 Actions effectuées en cas de reconfiguration.
13

14 Informations transmises en cours de tâche.

15 Récupération en fin de tâche.

16 Transmission de ressources en fin de tâche.

17 Transmission de ressources libérées en fin de tâche.

IVD FONCTION DE SURVEILLANCE.

Comme il a été écrit précédemment, la fonction de surveillance a d'une part un rôle préventif, d'autre part un rôle curatif. Cette section étudie particulièrement comment la fonction de surveillance détecte et localise une panne. Cette localisation consiste d'une part en une localisation géographique, d'autre part en une localisation hiérarchique (i.e. la connaissance, si possible, du (des) niveau(x) de machine(s) fautif(s)).

Dans ce but, il s'agit de tester les arcs du réseau de commutation ainsi que l'objet (terme qui recouvre les quatre niveaux supérieurs de machines) connecté à chaque cellule. La méthode générale est d'envoyer depuis un micro-processeur-tâche (M.P.T.) des interrogations et de gérer les réponses reçues, sachant que :

- une réponse correcte laisse augurer que les liaisons et composants ayant manipulé le message sont dans un état correct

- qu'une réponse incorrecte ou une absence de réponse (détectée par un mécanisme du type chien de garde) font supposer soit qu'une liaison ou un composant est en mauvais état, soit qu'une panne franche s'est produite.

Voyons le cas général (il est à mettre en oeuvre lors du démarrage de la machine pour tester l'état de ses constituants et lorsqu'une panne est supposée sans aucun a priori sur sa localisation).

IVD1 CONTRAINTES.

Les principales contraintes supportées dans le cas général sont :

- simplicité de l'algorithme.

Cette simplicité a pour buts une exécution rapide afin de satisfaire au critère de disponibilité du système, des échanges réduits afin de ne pas surcharger une machine dont l'état est présumé dégradé, un encombrement réduit en mémoire. Cet encombrement réduit doit d'une part concerner le micro-processeur qui teste afin qu'il stocke (au moins au départ) les informations dans sa mémoire locale, d'autre part concerner les composants (et en particulier les micro-processeurs) testés afin que la routine à effectuer en cas de test soit simple et peu encombrante à stocker en mémoire morte (ou à câbler).

- indépendance.

L'algorithme doit être indépendant de la position du micro-processeur qui effectue le test dans le réseau.

- extensibilité.

L'algorithme doit rester valide en cas d'extension du réseau ou en cas d'ajout de composants multiplexés sur une station d'échanges.

- totalité.

L'algorithme doit permettre de tester l'ensemble des constituants de la machine au moins une fois chacun.

- redondance.

Il est nécessaire que l'algorithme permette des vérifications redondantes (mais non identiques) afin d'une part de faciliter la localisation des pannes, d'autre part de permettre la poursuite du test bien qu'une panne ait été reconnue (ce n'est peut-être pas la seule).

- progressivité.

Elle consiste à ne vérifier le fonctionnement d'un ou plusieurs objets que si les antécédents (ensemble des objets transmettant le test vers cet ensemble) ont un fonctionnement déjà reconnu correct. La progressivité permet un algorithme plus simple à gérer et une localisation plus aisée.

IVD2 STRATEGIES GENERALES.

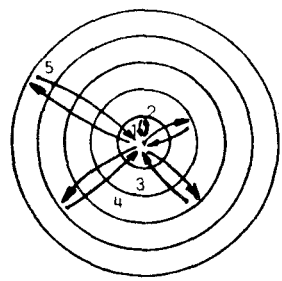
Parmi les stratégies possibles citons :

- le test par blocs.

Le système est découpé en blocs (ensembles de constituants). Dans une première phase le test est effectué à l'intérieur de chaque bloc ; dans une seconde phase les relations entre blocs sont testées.

- le test par nappes.

C'est le type même du test progressif. A partir d'un point, une première nappe de tests est effectuée dans le voisinage ; une fois les réponses reçues, une seconde nappe de tests est émise vers une couronne plus éloignée, puis une troisième après réception... Le test est complet lorsque le constituant le plus éloigné a été testé.



- le test par balayage.

Un seul message de test est émis. Deux possibilités sont à envisager :

- un seul retour. Le message parcourt l'ensemble du système et ce n'est à son retour (ou en abscence de retour) que des informations sont obtenues. Cette solution est très peu intéressante.

- retours multiples Ici encore le message parcourt l'ensemble du système mais des messages de réponses partielles sont générés (selon le principe de la chaîne décrit en IIIC6c) et renvoyés vers l'origine du test où ils sont gérés au fur et à mesure.
- etc...

Des combinaisons de ces stratégies de base peuvent être mises en oeuvre, par exemple test en nappes avec balayage à l'intérieur de chaque nappe ou test par balayage de blocs.

IVD3 EXEMPLE D'APPLICATION.

Cet exemple concerne le test par nappes. Il est mis en oeuvre en utilisant le principe de délégation décrit en IIIC3c. Le principe est que si un processeur Z effectue le test, après la nappe k les composants $\{A_x, A_{x+1},\ldots,A_y\}$ ont été reconnus comme des antécédents corrects ; la nappe k+1 du test consiste à effectuer le colloque de Z :

$$Z \left\{ \begin{array}{ccccccc} \overline{I} & A_x & \overline{D} & B_1 & \overline{D} & A_x & \overline{R} \\ \overline{I} & A_x & \overline{D} & B_2 & \overline{D} & A_x & \overline{R} \\ & & & & & & \\ \overline{I} & A_y & \overline{D} & B_n & \overline{D} & A_y & \overline{R} \end{array} \right\} Z$$

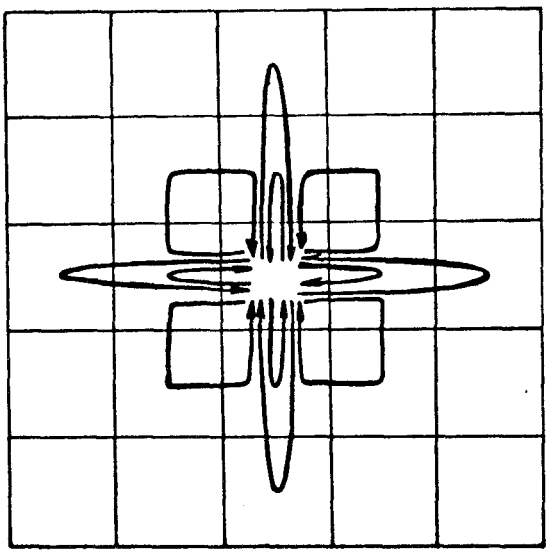
où $\{B_1, B_2,\ldots,B_n\}$ représente les constituants atteints par la nappe k+1. Le test concerne donc plus précisément les constituants qui effectuent :

$$A_x, \overline{D}\ B_i\ \overline{D}\ A_x,$$

Les interrogations de Z sont adressées aux A_x, (en fait, d'après le chapitre III ce sera à la station de communication de A_x, de déléguer) et de là aux B_i. Lorsqu'un constituant B_i reçoit un message de test, il doit effectuer une opération de test (ce qui contrôle le site B_i), gérer le protocole (ce qui contrôle le fonctionnement de la station de services) et renvoyer un message contenant le résultat de l'opération de test. Les autres constituants traversés (arcs du réseau entre A_x, et B_i, station de communication de B_i, station d'échanges de B_i, arcs du réseau entre B_i et A'_x) sont testés par la même occasion.

Grâce aux réponses qu'il reçoit (ou ne reçoit pas), Z peut calculer les antécédents corrects pour la nappe k+2 et entreprendre un nouveau colloque. La notion de nappes discontinues n'est d'ailleurs pas indispensable : Z peut lancer des tests dès qu'un nouvel antécédent correct a été déterminé.

La principale difficulté consiste à tester tous les arcs du réseau PROCOPE. Nous avons choisi, à partir de $A_x(I,J)$, d'effectuer les parcours suivants :



Si tous ces tests étaient effectués, il faudrait générer

$$12\,n^2 - 20n + 4 \quad (n \geq 5) \text{ messages.}$$

ce qui permet un test complet et redondant.

Certaines redondances peuvent être éliminées (d'où une notable diminution du nombre de messages) :

- les parcours tels que :

$$|\Delta I| = 1,\ |\Delta J| = 0 \text{ et } |\Delta I| = 0,\ |\Delta J| = 1$$

peuvent n'être effectués que sur la périphérie ($I=2$, $I=n-1$, $J=2$, $J=n-1$) et lors de la première nappe

- les parcours tels que :

$$|\Delta I| = 2,\ \Delta J = 0 \text{ et } \Delta I = 0,\ |\Delta J| = 2$$

peuvent n'être effectués que respectivement au Nord ($\Delta I = -2$), au Sud ($\Delta I = +2$), à l'Est ($\Delta J = +2$), à l'Ouest ($\Delta J = -2$) de Z (i,j) si respectivement $I < i-1$, $I > i+1$, $J > j+1$, $J < j-1$.

- les parcours tels que $|\Delta I| = |\Delta J| = 1$ peuvent n'être effectués que pour $I \geq i$ ($\Delta I = +1$, $\Delta J = \pm 1$) ou pour $I \leq i$ ($\Delta I = -1$, $\Delta J = \pm 1$).

Le test de l'ensemble du réseau a été simulé ainsi ; le tableau suivant regroupe quelques résultats.

	n=8	n=12
Origine (Z) : i, j	4,4	6,6
Nombre de messages	246	558
Nombre d'arcs non testés	0	0
Nombre de messages (avec redondances : $12\,n^2 - 20n + 4$)	612	1492

IVE FONCTION ALLOCATRICE.

La fonction allocatrice définie(IVC3d)a essentiellement pour rôle de rechercher, choisir, réserver les ressources nécessaires à la constitution d'une machine éphémère (M.E.). Partant des spécifications fournies par la fonction de gestion des tâches elle détermine donc la configuration d'exécution.

IVE1 ETATS DES RESSOURCES.

Un micro-processeur-tâche (M.P.T.) peut être dans les états suivants : en panne, au repos, affecté à la fonction utilisatrice, affecté à une fonction du système d'exploitation. S'il est affecté à une fonction utilisatrice, il peut être :

- chargé d'une tâche
- non chargé d'une tâche. C'est alors un objet alloué au micro-processeur chargé d'une tâche , il joue un rôle d'esclave [LEV73] vis-à-vis du maître qui en est chargé.

Les autres ressources peuvent selon leur nature et leur utilisation être dans un des états suivants : en panne, au repos, allouée, non allouée. Une unité non allouée est par exemple un micro-processeur fonctionnel mis à la disposition des machines éphémères ; l'accès s'y fait généralement grâce à un nom générique.

IVE2 CARACTERISATION DE L'ALLOCATION.

Soient les ensembles suivants à un instant t.

$\{M.E.\}_t$: ensemble des ressources allouées aux machines éphémères. Ces ressources ne sont pas préemptibles.

$\{T_i\}_t$: ensemble des tâches candidates à l'allocation (i.e. dont les ressources utilisateur sont réunies).

$\{A.\ T_i\}_t$: besoins de la tâche T_i appartenant à $\{T_i\}$ en ressources à allouer. Ces besoins peuvent être répartis en classes ; une classe peut être spécifiée par un nom générique (e.g. un micro-processeur-tâche, 2k octets de mémoire partageable) ou par une référence individuelle (e.g. l'imprimante X).

$\{F.\ T_i\}_t$: besoins de la tâche T_i appartenant à $\{T_i\}$ en ressources fonctionnelles (à ne pas allouer ; leur existence et leur disponibilité doivent être vérifiées).

$\{F.M.\}_t$: ensemble des ressources au repos gérées par la fonction magasinier.

Si l'on considère l'allocation indépendamment des autres fonctions (en particulier la fonction récupératrice), l'allocation peut être représentée par une fonction ψ :

$$\{F.M.\}_t\ ,\ \{T_i\}_t \overset{\psi}{\rightarrow} \{F.M.\}_{t+\Delta t}\ ,\ \{T_j\}_{t+\Delta t}\ ,\ \{M.E.\}_{\Delta t}$$

où $\{F.M\}_{t+\Delta t}$ représente les ressources résiduelles, où $\{T_j\}_{t+\Delta t}$ représente les tâches résiduelles (allocations impossibles), où $\{M.E.\}_{\Delta t}$ représente les nouvelles machines éphémères déterminées durant l'intervalle Δt.

La fonction ψ est caractérisée d'une part par les algorithmes mis en oeuvre, d'autre part par :

- le nombre d'objets allocateurs
- la connaissance de $\{F.M.\}_t$ et $\{T_i\}_t$.

IVE3 CHOIX DE L'ALLOCATION.

Les caractéristiques de la fonction ψ énoncées ci-dessus nous semblent traduire des choix plus fondamentaux portant en particulier sur les éléments suivants :

- la structure de la fonction magasinier
- la transmission du droit pour un objet de participer activement à la fonction allocatrice.
- la méthode de recherche des ressources par un composant de la fonction allocatrice

Les choix parmi diverses stratégies possibles doivent respecter les contraintes énumérées en IVC2 qui sont essentiellement :

- la recherche de la décentralisation
- la recherche de la souplesse et de la sécurité
- la limitation de la surcharge de travail causée par la fonction allocatrice et les fonctions associées.

IVE3a Structure de la fonction magasinier.

Le rôle de la fonction magasinier recouvre deux aspects :

- spécification de l'état de disponibilité de chaque ressource
- spécification la plus simple possible de l'ensemble des ressources d'une même classe dont l'état est : disponible.

Le second aspect n'est pas indispensable si l'on accepte d'interroger systématiquement l'état de toutes les ressources ; il permet cependant d'accélérer et simplifier notablement cette recherche.

Sur la plupart des machines, la fonction magasinier est centralisée dans des tables contenant l'état des ressources du système ; cette solution a été jugée peu désirable et les solutions suivantes lui sont préférables.

1) Magasinier circulant.

La fonction magasinier est dite circulante lorsqu'un "courrier" circule en permanence entre les objets qui peuvent exécuter la fonction allocatrice. Ce courrier contient la liste des ressources disponibles (i.e. qui peuvent être allouées) et un objet allocateur y prend des ressources (qui ont été restituées par un objet récupérateur). Cette solution est centralisée (mais non localisée) et donc peu fiable ; elle nécessite en permanence une circulation du courrier mais rend aisée la recherche des ressources disponibles. La liste des ressources disponibles doit être unique (sinon la mise à jour en est complexe) et donc le courrier est unique ou fragmenté en plusieurs courriers concernant différentes classes disjointes de ressources. En conséquence, le nombre d'objets allocateurs à tout instant est soit limité à un, soit limité à un par classe de ressources. La fragmentation en classes de ressources peut toutefois poser de délicats problèmes d'étreinte fatale ("deadlocks") par réservation de ressources dans certaines classes alors que les besoins ne sont pas satisfaits dans les autres classes.

2) Magasinier réparti.

La fonction magasinier est dite répartie lorsque chaque ressource est accompagnée de la spécification de son état et que cette spécification est accessible par la fonction allocatrice.

Cette solution est décentralisée, souple, mais sa répartition peut entraîner des communications nombreuses et délicates à gérer lors de la recherche de ressources par la fonction allocatrice. Si elle satisfait bien le premier aspect cité, elle satisfait moins bien le second et il faut la structurer afin de faciliter la recherche. Dans certains cas, cette structuration est simple (par exemple lorsque la machine présente une régularité géométrique) dans d'autres elle est implicite ; elle peut également être explicitée par le jeu de pointeurs, chaque objet contenant un pointeur vers une autre ressource équivalente. Selon la mise en oeuvre la liste obtenue contient soit toutes les ressources de

la classe, soit seulement celles qui sont libres. (Cette dernière option oblige à des mises à jour fréquentes et pose le problème de la liste vide).

IVE3b Droit à la fonction allocatrice.

Le droit à la fonction allocatrice est le droit d'accéder au rôle d'allocateur par un objet (M.P.T.) du système. Trois principales solutions nous ont semblé envisageables dans une architecture multi-micro-processeur : le droit stocké, le droit circulant, le droit négocié.

1) Le droit stocké.

Le droit à la fonction allocatrice peut être stocké dans une mémoire accessible par tous les objets désirant le prendre. Les problèmes d'exclusion, de protection sont alors résolus par des mécanismes classiques (sémaphores, moniteurs, etc...) qui gèrent également une file d'attente des demandes.

2) Le droit circulant.

Comme le magasinier, le droit à la fonction allocatrice peut circuler sous forme d'un courrier entre les objets qui peuvent en avoir besoin. Lorsque ce droit est effectivement désiré, l'objet attent le passage du courrier, s'approprie le droit, l'utilise et renvoie un courrier une fois l'allocation terminée. Cette méthode simple garantit l'exclusion mutuelle. mais ne permet en principe qu'un seul objet allocateur au plus à tout instant.

Une variante est possible en fragmentant le droit en droits partiels (et donc en plusieurs courriers) valables chacun pour une classe de ressources. Cette variante présente le risque de goulot d'étranglement cité en IVE3a.

3) Le droit négocié.

Le droit à la fonction allocatrice est dit négocié lorsque son acquisition résulte d'une négociation entre l'objet demandeur et les autres objets du système. Cette négociation, pour séduisante qu'elle paraisse, est dangereuse. Si des critères de choix raisonnables existent (par exemple la date de la demande) pour résoudre les conflits, cette solution entraîne une grande lenteur de la décision (il faut toujours attendre l'avis ultime avant de prendre le droit), ralentit le fonctionnement de toutes les unités consultées, est très sensible aux pannes des unités (si une unité ne répond pas, que faire ?).

IVE3c Recherche de ressources.

La recherche de ressources par un objet allocateur (i.e. qui a acquis le droit) dépend de la structure de la fonction magasinier. Si le magasinier est réparti, la recherche se fait soit par interrogations "en étoile" vers toutes les ressources, soit par circulation d'une interrogation (si le magasinier est structuré en liste). Si le magasinier circule sous forme d'un courrier, la recherche est immédiate.

IVE3d Choix de l'allocation.

Le choix d'un mécanisme d'allocation doit tenir compte :

- des différentes possibilités énumérées dans les paragraphes précédents
- des combinaisons possibles ou non entre ces possibilités
- des besoins du système
- des contraintes de réalisation et des choix déjà effectués.

A propos des besoins, il faut déterminer si plusieurs allocateurs sont nécessaires à un même instant. D'après les choix posés au chapitre I sur les caractéristiques des transactions et des tâches, d'après le nombre de processeurs, un calcul simplifié montre que dans le système

envisagé un seul allocateur peut suffire.

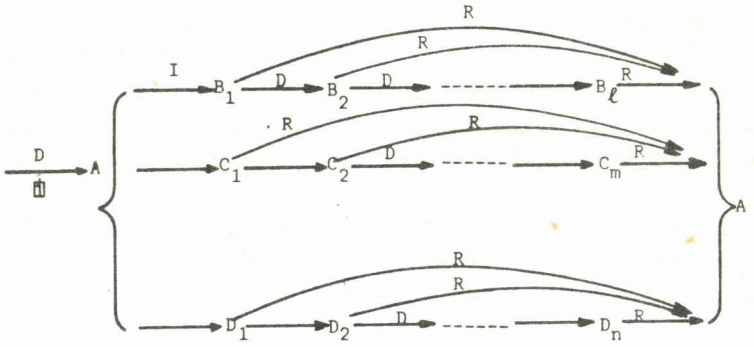
Soient des tâches de 1000 instructions exécutées sur des 8080 INTEL, leur durée avoisine 3000 µs ; soient 50 processeurs (M.P.T.) dans le système. Une approximation par la loi binômiale montre que l'espérance mathématique du nombre d'allocateurs est de 0,80 (ainsi que la variance). Si l'on se place dans des conditions moins draconiennes (tâches plus longues, ralentissements dûs aux communications, exécution des fonctions système, etc...) ce chiffre décroît ; afin de simplifier le système d'exploitation un seul allocateur au plus à tout instant paraît être le meilleur choix.

D'après toutes ces raisons, la fonction magasinier est à répartir dans le système ; pour accélérer les recherches elle est structurée en listes circulaires. Un objet d'une classe C_i possède un pointeur vers un autre objet de cette même classe, ces pointeurs sont mis en oeuvre grâce aux noms génériques vus au chapitre précédent (cf. IIID5) et sont stockés dans la mémoire M.S.E. . La liste doit être crée au démarrage du système, structurée en fonction d'un critère de proximité entre les objets, mise à jour en cas de panne ou de reconfiguration. Lorsqu'un objet allocateur recherche des ressources, il entamme un colloque (une consultation par type de ressource). Les consultations sont des chaînes (cf. IIIC6c) et donc il reçoit des réponses des objets disponibles au fur et à mesure que la liste est balayée ; ceci permet d'accélérer l'allocation si l'on admet non pas de rechercher la meilleure (du point de vue de la distance, des performances) ressource mais une bonne ressource (i.e. satisfaisant un minimum de contraintes). Au cas où la mémoire M.S.E. d'un objet pouvant être allocateur ne peut contenir tous les couples (nom générique - référence physique), il peut être fait appel à une mémoire externe contenant tous les couples.

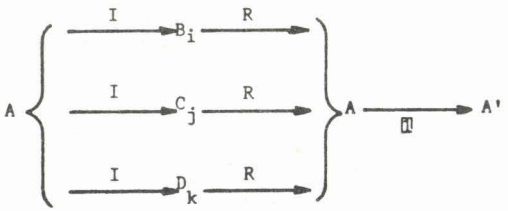
Le droit à la fonction allocatrice est un droit circulant sous forme d'un courrier entre tous les objets pouvant être allocateurs (M.P.T.). La circulation se fait grâce à la liste circulaire des micro-processeurs-tâches (M.P.T.). Afin d'augmenter la sécurité de fonctionnnement, un processeur désirant acquérir ce droit peut enquêter sur sa localisation

(pour vérifier son existence) et demander le courrier directement sans suivre la liste pré-établie.

La conférence d'allocation peut par exemple prendre la forme suivante.



Durant cette phase où ⊡ représente le courrier véhiculant le droit à la fonction allocatrice, l'objet A mène une enquête afin de connaître l'état des ressources dont il pourrait avoir besoin. Après avoir choisi les ressources, la conférence se poursuit par des réservations de ressource et le renvoi du courrier.



Dès que l'une des chaînes du colloque de recherche donne un résultat négatif (pas de ressource libre dans une classe), la conférence peut être terminée par renvoi du courrier. Les identificateurs restants sont gérés ainsi qu'il a été indiqué en IIIC7b.

CONCLUSION

Ce travail a tenté de définir une architecture parallèle d'ordinateurs adaptée à un type particulier d'application. Le choix de la gestion transactionnelle et l'étude de ses caractéristiques ainsi que des considérations technologiques et économiques ont mené à une première définition d'architecture multi-micro-processeur (ch. I).

L'accent a ensuite été mis sur les problèmes de communications, aspect essentiel des architectures multiprocesseurs. Dans une première étape (Ch. II) le réseau cellulaire proposé par V. Cordonnier et R. Cousin a été étudié en détail ; des modifications y ont été apportées afin de spécifier un outil opérationnel (qui sera réalisé dans un proche avenir) : le réseau PROCOPE. Dans une seconde étape (Ch. III), les communications ont été abordées d'un point de vue logique. D'une part, une hiérarchie fonctionnelle de machines a été établie ; d'autre part les protocoles de communication entre unités ont été étudiés afin d'obtenir une structure riche, souple et décentralisée.

Dans un quatrième chapitre, les outils définis antérieurement ont été utilisés afin de préciser l'architecture et le système d'exploitation envisagés. Cette définition est certes incomplète puisque bien des aspects n'ont été qu'effleurés ; elle demande également à être confrontée à la réalité (où a un simulacre de réalité) afin de la valider. Elle nous semble toutefois pragmatique et cohérente, en particulier en ce qui concerne les communications. Les outils de communication définis peuvent d'ailleurs être utilisés dans d'autres études, tant ce qui concerne le réseau cellulaire (processeurs tableaux, processeurs associatifs, etc...) qu'en ce qui concerne les protocoles (mise en oeuvre dans des réseau locaux de mini, voire micro, ordinateurs).

BIBLIOGRAPHIE

[ALE74] *Memory requirements for future computer systems.*
C.P.J. Alewijnse
L'Onde Electrique. 1974 Vol.54 n°6

[AN62] *D-825 a multiple-computer system for command and control.*
J.P. Anderson, S.A. Hoffman, J. Shifman, R.J. Williams
AFIPS Proc. FJCC Vol.22 1962 (repris dans CSRE (**))

[ARD75] *A multi-microprocessor computer system architecture.*
B.W. Arden, A.D. Berenbaum
Proceedings 5th. Symposium on Operating system principles.
1975

[ARN77] *Communication protocols in a multiprocessor system.*
A. Arnold, B. Degrendel
Publication du Laboratoire d'Informatique - Université des Sciences et Techniques de Lille

[BAE] *Preparation and evaluation of computer programs for parallel processing systems.*
J.L. Baer, E.C. Russel
PPSTA (*)

[BEA76] *Affectation quadratique.*
C. Beaudelot-Roucairol
Doctorat de troisième cycle. Octobre 1976. Université de Paris VI

[BEL] *The PMS and ISP descriptive systems.*
C.G. Bell, A. Newell
CSRE (**)

[BER66] *Analysis of programs for parallel processing.*
A.J. Bernstein
I.E.E.E. Transactions on Electronic Computers. Vol.EC-15 n°5 Oct. 66

[BLE76] *Aspects of a dynamically adaptative operating system.*
Parker R. Blevins, C.V. Rammoorthy
I.E.E.E. Transactions on Electronic Computers. Vol.C-25 n°7 July 76

[BOR76] *Gestion transactionnelle, un atout pour les minis.*
Luc Borer
Minis et micros n°33 Septembre 1976

[BOW76] *The modeling and design of multiple function-unit processors.*
J.W. Bowra, H.C. Torng
I.E.E.E. Transactions on Electronic Computers. Vol.C-25 n°3. March 76

[BRO69] *Mass memory in computer systems*
F.P. Brooks
I.E.E.E. Transactions on Magnetics. Vol.Mag-5 n°3 September 69

[CAM] *The specification of process synchronization by path expressions*
R.H. Campbell
Colloque IRIA

[CHU75] *Four-chip microprocessor family reduces system parts counts.*
D. Chung
Electronics, March 6, 1975

[COR75] *A two dimensions pipe-lined processor for communication in a parallel system.*
V. Cordonnier
SCCPP(*)

[COR77] *Un processeur de communications pipe-line.*
V. Cordonnier, R. Cousin
Digital Process (à paraître)

[COU76] *Etude et définition d'un processeur cellulaire de communications à haut degré de parallélisme.*
R.H. Cousin
Thèse de doctorat de spécialité. Université des Sciences et Techniques de Lille. Janvier 1976

[DAC72] *L'art et la manière d'accomoder une thèse.*
Pierre Dac
Les Pensées
1972 Editions Saint-Germain-des-Près

[DAR74] *Architectures de calculateurs orientés vers les systèmes.*
P. Darondeau, S. Guiboud-Ribaud, C. Otrage, H. Smit
R.A.I.R.O. a) juin 1974
b) janvier 1975
c) mai 1976

[DEN66] *Programming semantics for multiprogrammed computations.*
J.B. Dennis, E.C. Van Horn
C.A.C.M. Vol.9 n°3 March 1966

[ENS75] *Multiprocessor architecture - A survey.*
P. H. Enslow
SCCPP(***)

[FLY] *A multiple instruction stream processor with shared resources.*
M.J. Flynn
PPSTA (*)

[FLY70] *Detection and parallel execution of independent instructions.*
G.S. Tjaden, M.J. Flynn
I.E.E.E. Transactions on Electronic Computers. Vol.C-19 n°10 Oct.70

[FLY72] *Some computer organizations and their effectiveness.*
M.J. Flynn
I.E.E.E. Transactions on Electronic Computers. Vol.C-21 n°9

[FUL73] *Some observations on semiconductor technology and the architecture of large digital modules.*
S.H. Fuller, D.P. Siewiorek
Computer, october 1973

[GRO73] *Characteristics of a general purpose transaction processing operating system.*
Dr. J.M. Gross
IWOCAG (****)

[HAN72] *Structured multiprogramming*
P.B. Hansen
C.A.C.M. Vol. 15 n°7 1972

[HANS72] *A comparison of two synchronizing concepts.*
P.B. Hansen
Acta Informatica 1, p. 190-199. 1972

[HIN72] *Control Data STAR-100 processor design.*
R.G. Hintz, D.F. Tate
Compcon 1972

[HOA74] *Monitors : an operating systems structuring concept.*
C.A.R. Hoare
C.A.C.M. Vol.17 n°10 October 1974

[HOB] *Survey of parallel processor. Approaches and techniques.*
L.C. Hobbs, D.J. Theis
PPSTA (*)

[HOL59] *A universal computer capable of executing an arbitrary number of subprograms simultaneously.*
J. Holland
Proceedings of the Eastern Joint Computer Conference 1959

[INT] *Reality 20. Computer system reference manual.*
Documentation Intertechnique.

[INT76] *Intel Data Catalog.*
1976

[JAC75] *Parallel processing in software and hardware. The Mascot approach.*
K. Jackson, C.I. Moir.
SCCPP (***)

[KLE76] *On communications and networks.*
L. Kleinrock
I.E.E.E. Transactions on Electronic Computers. Vol.C-25 n°12 Dec.76

[KOC69] *The design of a highly parallel computer organisation.*
L.J. Koczela, G.Y. Wang
I.E.E.E. Transactions on Electronic Computers. June 1969

[LEG75] *Les protocoles de transport dans les réseaux à commutation par paquets : présentation et évaluation.*
H. Le Goff, R. Pedrono
Symposium technique du Forum mondial des Télécommunications. Genève Octobre 1975.

[LEL74] *Réseaux informatiques.*
G. Le Lann
Université de Rennes 1974

[LEL75] *Operating principles for a distributed multi-microprocessor.*
G. Le Lann, R. Négaret
Euromicro Workshop Nice 1975

[LEV73] *Computing with multiple microprocessors.*
J.V. Levy
IWOCAG (****)

[MAC70] *An introduction to the ILLIAC IV computer.*
D. Mc Intyre
Datamation April 1970

[MAI73] *Influence of technology evolution on system architecture*
F.P. Maison
IWOCAG (****)

[MET76] *Ethernet : Distributed packet switching for local computer networks.*
R.M. Metcalfe, D.R. Boggs
C.A.C.M. July 1976

[MIL73] *A comparison of some theoretical models of parallel computation.*
R.E. Miller
I.E.E.E. Transactions on Electronic Computers. Vol.C-22 n°8
August 73

[NOE73] *Macro-E-nets for representation of parallel systems.*
J.D. Noe, G.J. Nutt
I.E.E.E. Transactions on Electronic Computers. Vol.C-22 n°8
August 73

[PAT75] *Micro-control for parallel asynchronous computers.*
S.S. Patil
Euromicro Workshop Nice 1975

[POP74] *Protection structures.*
G.J. Popek
Computer, June 1974

[ROG74] *Present state and evolution of mass memories.*
H. Rogge
L'Onde Electrique 1974 Vol.54 n°6

[RUM75] *Data flow languages.*
J.M. Rumbaugh
SCCPP (***)

[SCH76] *On the efficient implementation of conditional critical regions and the construction of monitors.*
H.A. Schmid
Acta Informatica 6 1976

[SEA75] *Microprocessor application in multiple processor systems.*
B.C. Searle, D.E. Freberg
Computer October 1975

[SYM] *Symbad. Traitement de transactions.*
Documentation C.I.I.

[VAL76] *The FORUM project. Network conferencing and its future applications.*
J. Vallée
Computer Networks n°1, 1976

[WAG66] *On connecting modules together uniformly to form a modular computer.*
E.E. Wagner
I.E.E.E. Transactions on Electronic Computers Vol. EC-15 n°6 1966

[WEN75] *The impact of electronic disks on system architecture.*
J.H. Wensley
Computer February 1975

(*) Parallel Processor Systems, technologies and applications
Edited by : L.C. Hobbs, D.J. Theis, J. Trumble, H. Titus, I. Highberg
SPARTANT Books New-York

(**) Computer structures : readings and examples
Edited by : C.G. Bell and A. Newell
Mc Graw-Hill Book Company

(***) Proceedings of the SAGAMORE Computer Conference on Parallel Processing 1975

(****) Proceedings of the International Workshop on Computer Architecture
Grenoble 1973

BIBLIOTHEQUE UNIVERSITAIRE
Section de SCIENCES
LILLE