

T
Numéro d'ordre :

gen 2010 5653

50376
1995
307
1

Année : 1995

50376
1995
307
1

THESE

présentée en vue de l'obtention du

**Doctorat de l'Université
des Sciences et Technologies**

de Lille

Spécialité : Informatique

par

FREDERIC HOOGSTOEL

**UNE APPROCHE ORGANISATIONNELLE DU TRAVAIL COOPERATIF
ASSISTE PAR ORDINATEUR.**

APPLICATION AU PROJET CO-LEARN

Jury :

Président

M. Bernard TOURSEL

Directeur

M. Alain DERYCKE

Rapporteurs

M. Michel BEAUDOIN-LAFON

M. Jacques KOULOUMDJIAN

Examineurs

M. Jean-Marc GEIB

M. Joaquim Arnaldo MARTINS



REMERCIEMENTS

Mes premières pensées vont naturellement aux membres de ma famille. Je tiens à les remercier pour leur affection, leurs conseils et leur soutien dévoué.

Je remercie Monsieur Jacques Kouloumdjian, Professeur de l'Université Claude Bernard de Lyon et Monsieur Beaudoin-Lafon, professeur de l'Université d'Orsay d'avoir accepté d'être les rapporteurs de mon mémoire de thèse.

Je tiens à remercier mon directeur de thèse, le professeur Alain Derycke, qui m'a fait confiance et m'a apporté les moyens et les conseils nécessaires à mes travaux de recherche.

Cette thèse n'aurait pu exister sans la collaboration amicale des membres des équipes NOCE et OPEN du laboratoire TRIGONE. En particulier, je tiens à témoigner ici toute ma sympathie à Pascal Croisy et Claude Viéville, de l'équipe NOCE, avec lesquels j'ai travaillé en étroite collaboration dans le cadre du projet Co-Learn. Merci également à Danièle Clément de l'équipe NOCE et aux membres de l'équipe OPEN, qui m'ont aidé à acquérir les connaissances nécessaires dans le domaine des sciences humaines et de l'apprentissage coopératif à distance.

L'équipe NOCE est un groupe qui doit en partie sa cohésion et l'implication de ses membres au dévouement de sa secrétaire, Isabelle Logez : qu'elle en soit remerciée.

Je tiens à remercier les membres du consortium Co-Learn pour les discussions que nous avons eues et qui m'ont permis d'enrichir mon approche de la problématique grâce à des éclairages culturels différents.

Pour terminer, je veux remercier les organismes qui ont soutenu mes travaux sur le plan financier : le Centre Université Economie d'Education Permanente, le Centre National de la Recherche Scientifique, la région Nord-Pas-de-Calais et l'Ecole Universitaire Des Ingénieurs de Lille.

TABLE DES MATIERES

	Page
INTRODUCTION	1
PARTIE I. PROBLEMATIQUE : ASSISTER LA COLLABORATION	5
CHAPITRE 1 LES ENJEUX DU SUPPORT DE LA COLLABORATION AU SEIN D'UN GROUPE	7
1. Qu'est-ce qu'un groupe ?	7
2. Coopération ou Collaboration ?	9
3. Analyse des enjeux du support de la collaboration par l'informatique	11
PARTIE II. EVALUATION DE LA CONTRIBUTION AUX ENJEUX DES LOGICIELS D'AIDE A LA COOPERATION	27
CHAPITRE 2 LES LOGICIELS SPECIFIQUES	29
1. SYSTEMES ORIENTES INTERFACE	29
2. SYSTEMES ORIENTES COMMUNICATION	36
3. SYSTEMES ORIENTES INFORMATION	42
CHAPITRE 3 LES SYSTEMES INTEGRATEURS	45
1. Les systèmes exploitant la métaphore de salle virtuelle.....	45
2. Les environnements de développement	49
CHAPITRE 4 SYNTHÈSE DE L'EVALUATION DE LA SATISFACTION DES ENJEUX	54

PARTIE III. LES DISCIPLINES DE RECHERCHE EN INFORMATIQUE FONDATRICES DE NOTRE ENVIRONNEMENT GLOBAL DE COOPERATION	61
--	-----------

CHAPITRE 5 DU SUPPORT DU PARTAGE D'INFORMATION AU SUPPORT DE LA COLLABORATION	64
--	-----------

1. Les systèmes multi-bases	64
2. Les bases de données coopératives	67
3. Base à objets ou système à objets persistants réparti : quel choix pour les premières plates-formes de collaboration ?	78
4. Résumé.....	81

CHAPITRE 6 DE LA COMMUNICATION ASSISTEE PAR ORDINATEUR A LA COLLABORATION INFORMATISEE	82
---	-----------

1. L'échange privé d'information	82
2. La communication et la répétition individuelles de l'information.....	82
3. Le tri individuel de l'information	83
4. Le filtrage individuel de l'information reçue.	83
5. La diffusion de l'information vers des groupes.	83
6. Le partage de l'information par une communauté.	84
7. Le partage d'une base de messages par un groupe.....	84
8. La circulation de l'information dans une organisation.	85
9. La conversation de groupe : une activité du groupe.....	86

CHAPITRE 7 LA COOPERATION: UNE QUESTION D'AGENTS	89
---	-----------

1. Des agents intelligents ouverts au monde	89
2. Les agents : des fantômes d'utilisateur dans la machine	91
3. L'agent : un élément clé pour l'adaptation des interfaces homme-machine à la coopération.....	92
4. Les agents communiquent en groupe	93
5. La métaphore d'Agent dans Co-Learn	95

PARTIE IV ANALYSE, CONCEPTION ET DEVELOPPEMENT DE CO-LEARN	96
CHAPITRE 8 SPECIFICATIONS : DEVELOPPER UN RESEAU EXPERIMENTAL D'APPRENTISSAGE COOPERATIF A DISTANCE	100
1. Les spécifications informelles : un environnement pour l'enseignement ouvert.....	100
2. Prise en compte des contraintes	104
3. Présentation générale du système proposé : Co-Learn.....	105
4. La métaphore de la Salle Virtuelle ou Room	107
5. La métaphore d'Outil et le paradigme d'Activité.....	109
CHAPITRE 9 LE CADRE DE L'ANALYSE	110
1. Les méthodes d'analyse considérées dans notre étude.....	110
2. Articulation des approches	119
3. Notre proposition : un environnement de conception, de développement et d'intégration d'activités de coopération.....	132
CHAPITRE 10 LES POINTS DE VUE DE L'ORGANISATION	133
1. La base de la pyramide : les situations d'apprentissage coopératif.....	134
2. Le deuxième niveau de la pyramide : la gestion des unités organisationnelles.....	179
3. Le sommet de la pyramide : la gestion générale de l'organisation	181
4. Point de vue de l'organisation de Ellis : les vues offertes a l'utilisateur	183
CHAPITRE 11 LES POINTS DE VUE DE L'INFORMATION	188
1. Le modèle ontologique.....	188
2. Le modèle conceptuel des données	193
CHAPITRE 12 LES POINTS DE VUE DYNAMIQUES	207
1. De l'application interactive personnelle au système d'information coopératif.....	207
2. Architecture logicielle distribuée de l'environnement de coopération Co-Learn.....	217

CHAPITRE 13 LES POINTS DE VUE DE L'INGENIERIE 226

- 1. Programmation des composants du système distribué Co-Learn 226
- 2. Techniques de contrôle logiciel..... 227
- 3. Techniques de Communication..... 228
- 4. Techniques de gestion de la persistance..... 232

CHAPITRE 14 LES POINTS DE VUE TECHNOLOGIQUES 235

- 1. Les serveurs Co-Learn..... 235
- 2. Les clients Co-Learn 236
- 3. Le réseau Co-Learn 237

PARTIE V EVALUATION ET AMELIORATION DE CO-LEARN 240

CHAPITRE 15 EVALUATION DE CO-LEARN ET PERSPECTIVES D'EVOLUTION 242

- 1. De l'ACAO au TCAO : développer le support des activités instrumentales 242
- 2. Augmenter la malléabilité de Co-Learn 254
- 3. Augmenter l'accessibilité du système Co-Learn..... 255
- 4. Mieux assister l'orientation dans l'environnement de coopération..... 256
- 5. Améliorer la gestion de la persistance..... 256

CONCLUSION 259

REFERENCES BIBLIOGRAPHIQUES 261

LISTE DES ILLUSTRATIONS 277

LISTE DES TABLEAUX 280

ANNEXE I : LE PROJET CO-LEARN	281
--------------------------------------	------------

ANNEXE II : LA BASE A OBJETS GEMSTONE	288
--	------------

ANNEXE III : CONCEPTION AMIGO D'UNE CONVERSATION DE PARTAGE DE RESSOURCE	294
---	------------

ANNEXE IV : CONCEPTION CO-LEARN D'UNE CONVERSATION DE PARTAGE DE RESSOURCE	307
---	------------

ANNEXE V : CONCEPTION CO-LEARN D'UNE ACTIVITE D'EDITION DE FRAGMENT COOPERATIVE	318
--	------------

INTRODUCTION

Les temps sont maintenant révolus où l'on concevait l'ordinateur comme une machine extraordinaire mais inaccessible capable de trier ou comptabiliser - parfois au prix de quelques heures ou quelques jours de travail - de longues listes de noms ou de chiffres. Cet engin était lui-même bien incapable de réutiliser les données qu'il nous restituait. Il nous laissait alors seul avec la charge d'exploiter cette information insipide, de la communiquer et la présenter au moment adéquat et sous la forme la mieux adaptée aux personnes que cela pourrait intéresser.

Aujourd'hui, l'ordinateur n'est plus, contrairement à ce que l'on peut croire, une machine à stocker, trier et calculer. C'est à la fois un outil ou un média de communication sophistiqué et un collaborateur efficace, disponible et communicant.

Aujourd'hui, l'ordinateur ne s'appelle plus ordinateur : appelons-le plutôt assistant personnel. Nous pouvons dire qu'il est à la fois ordinateur, fax, téléphone, visiophone et télévision interactive. C'est en fait, selon ce que nous désirons, un simple outil ou un agent intelligent à votre service.

Computer systems are coming to be used more like consultants or even collaborators, and less like tools

Zachary & Robertson, "Introduction to Cognition, Computation, and Cooperation" [ZACHA 90]

L'ordinateur n'est plus un outil de calcul isolé mais une entité semi-autonome d'un réseau d'agents coopérants, une porte ouverte sur un monde de connaissance, de communication et de collaboration.

Il est vrai que nous entrons dans l'ère d'une nouvelle écologie de l'informatique. La bureautique devient communicante et coopérante, les systèmes informatiques sont ouverts et distribués. La communication supportée par ordinateur devient coordination, conversation et collaboration informatisées. L'intelligence artificielle mime l'intelligence naturelle : elle se distribue parmi des agents coopérants. Les bases de données se fédèrent et coopèrent. Les interfaces Homme-Machine abandonnent le langage de commande et adoptent un langage de coopération : elles n'attendent plus que l'utilisateur parvienne à formuler ses requêtes dans le langage informatique mais essaient de le guider dans ses recherches. Elles offrent des moyens de méta-communication tels que la manipulation et la désignation directes ainsi que la

communication tels que la manipulation et la désignation directes ainsi que la communication multimodale, exploitant tous les moyens de communication de l'être humain. Mais les interfaces Homme-Machine savent aussi se faire discrètes quand il le faut, c'est à dire pour servir de média à la collaboration Homme-Homme.

Même les mécanismes internes de communication de l'ordinateur subissent une mutation qui les rendra bientôt plus adaptés à la communication de groupe qu'au dialogue.

Les concepteurs d'applications assistent et participent eux aussi à cette révolution : un programme n'est plus conçu de façon monolithique mais il est composé d'un ensemble de modules coopérants, d'objets actifs ou d'agents. Le génie logiciel propose les moyens de répondre à ses exigences qui ont toujours visé implicitement à favoriser la coopération : le programmeur n'écrit plus un programme pour lui seul mais conçoit un système d'information que lui-même et d'autres seront amenés à utiliser, améliorer, étendre, corriger... Les ateliers de génie logiciel multi-utilisateurs supportent la coordination des membres des groupes de conception ou de développement.

L'avènement du paradigme objet accompagne celui de la coopération. L'exécution d'un programme consiste en un échange incessant de messages entre objets. Les systèmes d'exploitation et d'interfaces deviennent événementiels, composés d'objets réactifs coopérants. L'utilisateur ne manipulera bientôt plus de fichiers et n'utilisera plus d'éditeurs : l'utilisateur peut déjà interagir directement avec les objets multimédia, hypermédia ou virtuels. Chaque objet réagit selon ses capacités pour participer de son mieux à cette coopération : désignez un son, il se fera entendre, désignez une séquence vidéo, elle se déroulera. Mieux, vous n'avez plus besoin de clavier : écrivez et l'ordinateur vous lit, parlez et l'ordinateur vous comprend, désignez et l'ordinateur voit les objets de votre conversation.

Les réseaux ne sont plus soit téléphoniques, soit informatiques mais numériques à intégration de services : l'ordinateur les exploite pour transmettre en même temps les objets de votre conversation, votre voix et vos gestes, en temps réel ou en différé. L'ordinateur sait utiliser ces réseaux pour créer un monde virtuel où tout le monde peut voir et entendre ce que l'autre perçoit, et coopérer en abolissant les barrières de temps et d'espace.

Les mondes des télécommunications, des médias, de l'édition et de l'informatique fusionnent pour prendre part à cette avancée de l'univers de

l'information qui augmente l'interactivité et la richesse de la communication pour promouvoir la coopération. Le besoin de coopération est unanime¹.

Dans un premier temps, il se fait particulièrement sensible dans le domaine de l'éducation, de la recherche et de la santé. Nous nous sommes particulièrement intéressés dans ces travaux à l'utilisation de l'informatique coopérative pour l'éducation. Le paradigme de la collaboration s'impose aussi dans notre société dans le domaine de l'éducation : l'heure n'est plus au cours magistral mais à l'apprentissage coopératif. On sait maintenant que l'on n'apprend pas seul ; il faut favoriser la construction sociale du savoir. Cette promotion de la collaboration est particulièrement importante dans le domaine de l'enseignement où l'on doit tout à la fois coopérer pour apprendre et apprendre à coopérer [SLAVI 85] [DERYC 93].

Cette thèse présente la participation de l'auteur aux travaux de recherche menés durant quatre années au sein de l'équipe NOCE (Nouveaux Outils pour la Communication Educative) du laboratoire Trigone, dans le domaine du Travail Coopératif Assisté par Ordinateur (T.C.A.O.). La recherche dans le champ du T.C.A.O. implique un travail en équipe pluridisciplinaire : c'est ce qui fait sa complexité mais aussi son intérêt. Il nécessite la coopération de plusieurs domaines de la recherche en informatique : Interfaces Homme-Machine, Systèmes de Communication, Réseaux, Systèmes de Gestion de Base de Données, Systèmes Distribués, Systèmes d'Information... Il demande aussi la collaboration des chercheurs en informatique avec les chercheurs en sciences humaines : Sociologie, Psychologie...

L'auteur a travaillé en étroite collaboration avec Pascal Croisy, dont les travaux sont présentés dans [CROIS 95] et qui s'est particulièrement intéressé à l'architecture des logiciels interactifs supportant la coopération simultanée, et Claude Viéville qui a essentiellement consacré ses travaux à la conception de logiciels de communication en temps différé. Les travaux de l'auteur visent à concevoir un environnement global d'analyse, de conception, de développement, d'intégration et d'administration des applications informatiques d'aide à la coopération, telles que celles étudiées par Pascal Croisy et Claude Viéville.

La première partie de ce mémoire présente la problématique de notre recherche. Nous la concluons en définissant six enjeux pour le support du travail coopératif dans une organisation.

¹en dépit de cet enthousiasme, force est de constater que cet engouement pour la coopération ne se traduit pas aisément dans les faits. Comme le note A. Derycke dans [DERYC 94] *actuellement, la collaboration et la coopération restent malgré tout des formes de travail très exigeantes pour les individus.*

La deuxième partie propose une étude pragmatique de la participation aux six enjeux des logiciels existants, qui montre le rôle que peuvent jouer les différents types de collecticiels dans un environnement intégré de TCAO.

La troisième partie montre les conséquences de la conception d'un environnement intégré de TCAO sur trois domaines de l'informatique : les systèmes de partage d'information (bases de données, hypermédia, systèmes distribués), les systèmes informatiques d'aide à la communication, les systèmes multi-agents.

La quatrième partie développe l'analyse et la conception de Co-Learn. Co-Learn est à la fois un système distribué opérationnel d'aide à l'apprentissage coopératif, un environnement de développement de nouvelles activités de coopération et une architecture d'accueil distribuée. Nous nous attachons à décrire les méthodes d'analyse que nous avons développées et mises en oeuvre pour développer l'environnement Co-Learn et qui sont utilisées pour concevoir et intégrer dans l'environnement de nouvelles activités de coopération.

Pour conclure, la cinquième partie consiste en une critique du système Co-Learn et analyse ce qui différencie un système d'aide à l'apprentissage coopératif tel que Co-Learn d'un système d'aide au travail coopératif. Nous étudions les différentes pistes qui pourraient être suivies pour obtenir un système de TCAO et leurs conséquences à la fois sur les méthodes d'analyse et de conception proposées par Co-Learn et sur l'architecture du système d'information distribué.

**PARTIE I
PROBLEMATIQUE : ASSISTER LA COLLABORATION**

Cette première partie a pour rôle de définir la problématique considérée dans cette thèse.

Elle reflète notre compréhension des concepts de groupe, de coopération et de collaboration, compréhension que nous avons développée par l'analyse de travaux pluridisciplinaires en sciences humaines et en informatique.

L'analyse que nous y développons mène à la définition de ce que nous considérons être les six enjeux du support de la collaboration dans une organisation.

Ces enjeux seront utilisés dans la deuxième partie pour étudier le rôle des systèmes existants d'aide au travail coopératif.

TABLE DES MATIERES DE LA PREMIERE PARTIE

**CHAPITRE 1 LES ENJEUX DU SUPPORT DE LA COLLABORATION
AU SEIN D'UN GROUPE** **7**

- 1. Qu'est-ce qu'un groupe ? 7
- 2. Coopération ou Collaboration ? 9
- 3. Analyse des enjeux du support de la collaboration par
l'informatique 11
 - 3.1. Favoriser la coordination du groupe 13
 - 3.2. Favoriser la communication dans le groupe 15
 - 3.3. Favoriser la cohésion du groupe 17
 - 3.4. Favoriser l'implication individuelle 18
 - 3.5. Favoriser le partage des ressources du groupe..... 19
 - 3.6. Faciliter l'organisation du groupe..... 20
 - 3.7. Récapitulation de la contribution des fonctionnalités aux
enjeux 23

CHAPITRE 1

LES ENJEUX¹ DU SUPPORT DE LA COLLABORATION AU SEIN D'UN GROUPE

1. QU'EST-CE QU'UN GROUPE ?

Les travaux d'Anzieu et Martin [ANZIE 82] constituent pour nous une référence précieuse en matière d'analyse des groupes, en particulier dans le domaine de l'analyse de la dynamique des groupes restreints. Nous rappelons ici les résultats de ces travaux et nous montrons comment nous proposons de les adapter pour prendre en compte l'apparition des nouvelles formes de groupes composés de personnes qui se réunissent dans des espaces virtuels.

Le mot de *groupe* est l'un des plus confus de la langue française. Il est récent dans l'histoire des langues : le concept objectif de *groupe*, fondement d'une science des associations, des comités et des équipes, a émergé lentement au cours de l'histoire de la pensée. Ce travail d'objectivation est gêné par des préjugés individuels et collectifs.

D'abord, la crainte de repenser sa propre situation dans un nouveau cadre de référence, et ainsi d'être remis en question, et les angoisses primitives (*angoisse de persécution, angoisse dépressive, angoisse de morcellement du corps, angoisse devant le désir d'une fusion symbiotique dans le groupe, anéantissante pour la personnalité individuelle*) constituent les aspects d'ordre psychologique et psychanalytique de la résistance épistémologique à la notion de *groupe*.

Ensuite, une résistance épistémologique d'ordre sociologique découle du *totalitarisme groupal* : *un groupe est fait pour être vécu totalement ; il n'est pas fait pour qu'on l'étudie, c'est à dire pour qu'un de ses membres prenne une distance par rapport à lui ou qu'un étranger s'y introduise par pure curiosité*. Une autre résistance est due à ce que, *pour la société globale, le groupe restreint est une force à son service, mais qui peut se retourner contre elle* (d'où, par exemple, la méfiance des professeurs à l'égard du travail en équipe).

¹L'analyse que nous présentons ici est complémentaire de celle que propose [CROIS 95], qui s'intéresse au support de la coopération en temps réel par des systèmes interactifs de groupe (collecticiels). Nous nous intéressons à l'aide à la coopération d'un groupe sur le long terme, qui donne lieu à de nombreuses situations de coopération dont les situations de coopération en temps réel étudiées par [CROIS 95].

De plus, le mot *groupe* est utilisé dans le langage courant au sens général d'ensemble de personnes. En 1982, Anzieu et Martin écrivaient qu'il serait souhaitable de réserver l'usage scientifique du vocable *groupe* à un *ensemble de personnes réunies ou qui peuvent et veulent se réunir*. Nous pourrions conserver cette définition dans cette étude, car nous voulons restreindre la notion de *groupe* dans une direction similaire. Toutefois, il faut noter que nous sous-entendons dans ce cas une autre définition du verbe *se réunir*. En effet, si le verbe *réunir* n'implique pas toujours la présence physique en un même lieu (il peut être utilisé dans le sens de *faire communiquer* [LAROU 81]), *se réunir* a toujours évoqué la nécessité de rassemblement en un même lieu à un même instant. Or l'objet de notre étude a justement commencé à effacer cette connotation, puisqu'elle doit permettre à un groupe de communiquer sans contraintes de temps ou de lieu. Il est important de remarquer que, bien qu'éliminant les contraintes de temps et de lieu, notre acceptation de *groupe* n'est pas moins restrictive que celle d'Anzieu et Martin. En effet, dans la définition proposée par Anzieu et Martin, réunir est utilisé dans le sens d'assembler en un même lieu. Nous utilisons réunir dans le sens de mettre en relation, faire communiquer alors que la définition d'Anzieu n'implique pas de communication ou de relation entre les membres du groupe. Ainsi, la définition d'Anzieu englobe la notion de foule, alors que notre définition l'exclue parce que les individus d'une foule ne communiquent pas, il y a seulement contagion des émotions. La foule est caractérisée par l'absence des contacts sociaux et des interactions inter humaines, la passivité des gens réunis envers tout ce qui n'est pas la satisfaction immédiate de leur motivation individuelle : la foule a la solitude en commun.

Pourtant, excepté la foule, les catégories de groupes identifiées par Anzieu et Martin répondent à notre définition du groupe, si l'on omet la condition de rassemblement en un même lieu, ou si l'on étend la notion de lieu à celle d'espace virtuel. Ces catégories sont la bande, le groupement, le groupe restreint et le groupe secondaire.

Notre but étant de supporter le travail coopératif, les catégories de groupe auxquelles nous nous intéressons se limitent au groupement, au groupe restreint et au groupe secondaire. En effet, les membres d'une bande ne se réunissent pas dans le but de coopérer à une tâche² organisée en commun : pour eux, se réunir est un but en soi.

²Le concept de tâche est utilisé pour traduire une intention opérationnelle indépendamment de la façon de la réaliser. Une Tâche est exprimée en terme de quoi, par opposition à une Activité, exprimée en terme de comment. Une tâche peut être accomplie, une activité peut cesser [SCHMI 93].

Les activités accomplies en commun n'apparaissent pas comme un but essentiel à la bande : le but est d'être ensemble parce que l'on est semblable.

Il est intéressant de noter les caractéristiques qui changent lorsque la bande devient durable en se transformant en un groupe primaire : elle affirme des valeurs communes, elle privilégie la loyauté et la solidarité de ses membres, différencie leurs rôles, fixe des buts autres que la complaisance collective à soi-même. Nous allons voir que ces nouvelles caractéristiques (valeurs communes, solidarité des membres, différenciation de rôles, fixation de buts) sont des conditions nécessaires à la collaboration.

2. COOPERATION OU COLLABORATION ?

L'ambiguïté de la coopération vient de celle du verbe coopérer. Cette ambiguïté apparaît dans les définitions que donnent le Larousse et le Robert de 1981 [LAROU 81] [ROBE 81].

D'abord, le Larousse définit coopérer uniquement comme un verbe transitif indirect, alors que le Robert lui reconnaît également une utilisation comme verbe intransitif. Cette différence est confirmée par une différence de sens : selon le Larousse, coopérer c'est *agir conjointement avec quelqu'un*; selon le Robert, c'est *opérer conjointement (avec quelqu'un)*. Coopérer est utilisé comme un verbe transitif indirect avec pour sens *agir conjointement avec quelqu'un* lorsque le sujet est une partie de l'ensemble des membres du groupe que l'on distingue des autres membres (quelqu'un) : chaque membre coopère au travail du groupe. Coopérer est utilisé comme verbe intransitif dans le sens *opérer conjointement* lorsque le sujet est l'ensemble des membres d'un groupe : le but n'est pas de dire que chaque membre apporte une contribution à une tâche (il n'y a pas de complément d'objet pour préciser quelle serait cette tâche) mais que les membres du groupe entretiennent des relations de coopération.

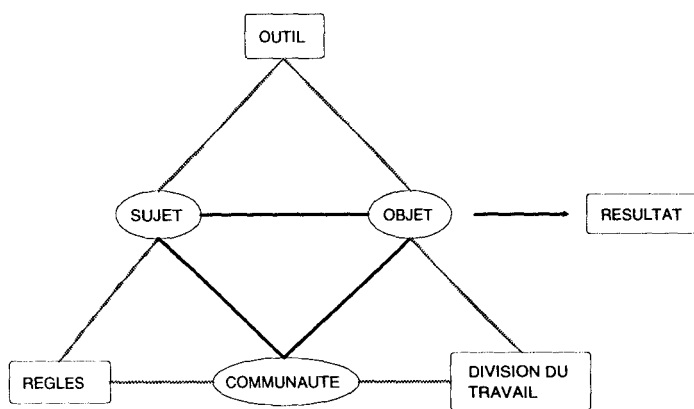
La coopération, parce qu'elle signifie action de coopérer, peut donc désigner l'activité menée par un membre du groupe pour réaliser la tâche commune (activité que nous désignerons par coopération individuelle) ou l'activité globale du groupe (que nous désignerons par coopération de groupe). Certains auteurs nient l'existence d'activité de groupe. De notre point de vue, on peut définir deux niveaux de coopération à assister. Le premier niveau est la coopération individuelle. Pour supporter ce niveau, il suffit de donner les moyens à chaque membre d'apporter sa contribution au travail entrepris par le groupe. Il faut noter qu'un tel niveau de coopération suppose simplement un apport, sans impliquer une participation complète

et voulue [LAROU 81]; en fait, l'individu ne fait aucun effort d'articulation de ses activités par rapport à celles des autres, ce niveau de coopération ne nécessite même pas l'existence d'un groupe. Le deuxième niveau est la coopération de groupe. Pour supporter ce niveau, il faut faciliter les relations de coopération entre les membres du groupe et, dans le cas d'une coopération à long terme, assurer la pérennité du groupe. La coopération de groupe est une activité sociale, le système informatique qui la supporte est l'élément d'intégration d'un milieu socio-technique. La coopération de groupe est plus que la coopération : c'est la collaboration, au sens où [LAROU 81] distingue collaborer et coopérer :

Collaborer (du lat. cum, avec, et laborare, travailler), c'est travailler de concert et de son plein gré, avec un autre, l'aider dans ses fonctions, dans ses efforts. *Coopérer* dit moins; il peut supposer simplement un apport, sans impliquer une participation complète et voulue.

[LAROU 81]

Nous pouvons mettre en évidence la différence qui existe entre coopérer et collaborer en considérant le modèle de l'activité proposé par [ENGES 87] et repris par [KUUTT 91] [KUUTT 93].



La structure de base d'une activité

Figure 1. Le modèle de l'activité de Engeström.

Dans la coopération, prise dans son sens le plus distinctif de la collaboration, il n'y aurait pas de relation entre le sujet et la communauté : le sujet n'aurait de relation qu'avec l'objet. A la limite, la communauté n'existerait pas : il n'y aurait plus qu'un ensemble d'individus isolés. Cependant, cette distinction n'est en général pas prise en compte et coopération et collaboration sont utilisés comme des synonymes.

3. ANALYSE DES ENJEUX DU SUPPORT DE LA COLLABORATION PAR L'INFORMATIQUE

Les premiers systèmes informatiques qui ont été conçus pour supporter la coopération n'avait pour ambition que de supporter techniquement des coopérations ponctuelles, à court terme. Le champ du **C.S.C.W. (Travail Coopératif³ Assisté par Ordinateur)**, dès lors qu'il s'intéresse au support de la coopération **pour le travail et donc au support d'une activité sociale**, ne peut plus se contenter d'assister les membres du groupe individuellement ou ponctuellement, il doit favoriser la collaboration dans la continuité, aider au développement de la mémoire, de la culture du groupe.

Notre but est donc d'utiliser l'informatique pour assister la coopération à ses deux niveaux : coopération individuelle et coopération de groupe (collaboration).

Permettre la coopération individuelle, c'est offrir à chacun les moyens d'accéder à l'espace des objets partagés par le groupe afin d'en consulter, d'en modifier ou d'en ajouter. Pour accéder à cet espace, on peut permettre à chacun d'utiliser des outils personnels ou/et offrir des outils partagés. La principale difficulté est de garantir l'intégrité de l'espace d'objets commun, ce que l'on peut résoudre par un contrôle de concurrence classique.

Faciliter la coopération de groupe et en particulier le travail coopératif, c'est en plus supporter une activité sociale.

Or, selon Anzieu et Martin, *toute activité sociale postule des échanges d'informations, soit entre les membres d'un même groupe, soit entre les membres de groupes différents : << Une société est faite d'individus et de groupes qui communiquent entre eux >>. De plus, il est nécessaire pour les groupes de s'organiser : en premier lieu, pour que soient collectées des informations utiles et efficaces; en second lieu, pour que ces informations soient distribuées convenablement entre tous ceux qui devront les utiliser, notamment ceux qui auront à les traiter pour prendre des décisions valides.*

Il apparaît donc que le travail coopératif, en tant qu'activité sociale implique et nécessite :

- la communication,
- l'organisation donc la coordination,
- la distribution d'information.

³L'adjectif coopératif n'a pas le sens restrictif de la coopération par rapport à la collaboration. Il faut noter que l'adjectif collaboratif n'existe pas. D'après [LAROU 81], coopératif signifie *qui participe volontiers à une action commune*. Etre coopératif sous-entends la "bonne volonté".

Dans leur analyse des communications dans les groupes restreints, Anzieu et Martin distinguent l'information et la communication.

L'information est à la fois : une opération (l'action d'informer) ; un contenu (ce qui informe), aboutissant à une réduction du désordre.

La communication est l'ensemble des processus physiques et psychologiques par lesquels s'effectue l'opération de mise en relation de une (ou plusieurs) personne(s) - l'émetteur - avec une (ou plusieurs) personne(s) - le récepteur -, en vue d'atteindre certains objectifs.

L'oubli de cette distinction a été une des causes des échecs des systèmes de support de la communication, comme nous le verrons dans le chapitre 2.

[ANZIE 82], citant Enriquez, distingue en outre :

- *la nature de la communication : processus d'affectation d'autrui;*
- *la fonction de la communication : contrôle et régulation des activités d'autrui.*

Anzieu et Martin analysent également les obstacles à la communication. Ils nous mettent d'abord en garde contre une approche uniquement technologique de la communication, qui n'identifie les problèmes qu'aux niveaux du codage, de la transmission, du décodage et de la rétroaction. Ils nous présentent à cet effet les aspects psychosociologiques de la communication, insistant sur le fait que lors d'une communication *entrent en contact deux ou plusieurs **personnalités** engagées dans une **situation commune** et qui se débattent avec des **significations**.*

On peut déduire de l'analyse qu'ils développent ensuite (Ibid., pages 190 à 200) que pour surmonter les obstacles, les participants d'une communication nécessiteront :

- d'avoir le choix des moyens riches d'expression et de formulation,
- de pouvoir faire converger les moyens (joindre le geste à la parole),
- de connaître la situation générale du groupe,
- de connaître le statut et le rôle des autres,
- de connaître clairement son propre rôle et son statut.

En conclusion, nous identifions six grands enjeux pour supporter le travail coopératif dans un groupe :

- Faciliter la **coordination** des participants,
- Faciliter la **communication** entre les membres,
- Favoriser la **cohésion** du groupe,
- Favoriser l'**implication** individuelle des membres,
- Faciliter le **partage** des ressources entre les participants.
- Faciliter l'**organisation** du groupe.

Le groupe de travail *SCOOP* du Pôle de Recherche Concertée sur la Communication Homme-Machine et l'atelier *Collecticiel* des journées IHM'94 ont retenu trois espaces dans lesquels interagissent les utilisateurs d'une application interactive partagée : l'espace de communication, l'espace de coordination et l'espace d'information (figure 2). Assister l'interaction des utilisateurs dans ces trois espaces constitue une partie de trois de nos enjeux, respectivement l'enjeu de communication, l'enjeu de coordination et l'enjeu de partage. Les enjeux de cohésion, d'implication et d'organisation apparaissent lorsque le groupe s'agrandit et s'organise.

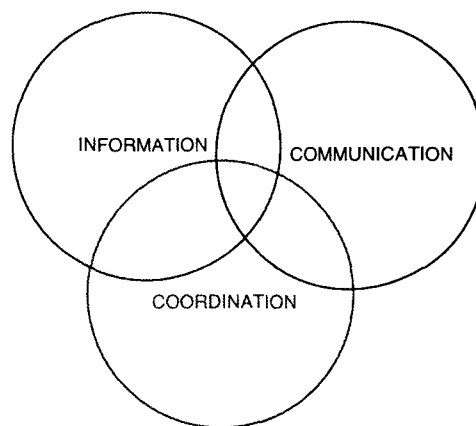


Figure 2. Le trèfle SCOOP

3.1. Favoriser la coordination⁴ du groupe

Pour se coordonner, ne serait-ce qu'implicitement⁵, les participants ont d'abord besoin de suivre l'activité des autres participants et l'utilisation et l'évolution des ressources partagées.

Mais il ne suffit pas de savoir ce que font les autres pour agir de manière coordonnée : il faut encore connaître et faire connaître le rôle que l'on peut jouer dans les activités du groupe. Il est donc essentiel de permettre de distribuer les rôles adéquats aux différents participants dans les différentes activités du groupe. Ces rôles ne doivent pas être statiques mais au contraire dynamiques pour faire évoluer les droits et devoirs associés à ces rôles en fonction de l'évolution de l'activité du groupe.

⁴Depuis plusieurs années émerge un corpus théorique multidisciplinaire de Sciences de la Coordination. [MALON 90] propose d'utiliser la Théorie de la Coordination pour aider à la conception des systèmes pour le travail coopératif. L'Articulation du travail coopératif considéré par [SCHMI 94] est synonyme de la coordination que nous considérons ici.

⁵La coordination implicite est celle que l'on rencontre lorsque la coopération est relayée structurellement, c'est à dire par l'espace d'action : c'est ce que [SCHMI 94] appelle *cooperative work mediated by the field of work*.

Pour que le groupe puisse discuter de l'attribution des tâches et des rôles, en explicitant la prise d'engagements, le système facilitera également la coordination explicite. Il s'agit bien sûr de mémoriser les engagements pris par les différents participants et de les leur rappeler. Mais il s'agit également de supporter l'activité de groupe de la prise d'engagement. En effet, lorsque les participants n'ont pas le loisir de se réunir, ils doivent communiquer par envoi de messages, ce qui complique le suivi des négociations par le groupe.

Le support de la coordination doit permettre aux membres d'un groupe de travailler pour le groupe dans différents modes de coopération, en présence comme en l'absence du groupe. Le système informatique doit tenir chaque membre au courant de l'évolution de la situation de coordination dans laquelle il est en train de travailler et lui offrir toujours les fonctionnalités adaptées à cette situation. Considérant l'environnement d'édition coopérative d'hypermédia SEPIA, Haake & Haake [HAAKE 92] [HAAKE 93] distinguent quatre situations de coopération différentes d'accès à l'espace d'information partagé qu'un système peut être amené à gérer :

- **la situation isolée**, dans laquelle un participant est seul à travailler sur une partie de l'espace d'information partagé ou à apporter sa contribution à une activité de conversation asynchrone (rédiger un message pour le groupe) ;

- **la situation séparée**, dans laquelle plusieurs participants préparent chacun une version différente de la même partie de l'espace partagé, sans interaction entre les participants;

- **la situation faiblement couplée**, dans laquelle les participants, bien qu'accédant concurremment à la même partie de l'espace partagé, peuvent travailler de façon indépendante sans autre moyen de communication et de coordination que la communication des effets des actions des autres participants ;

- **la situation fortement couplée**, qui permet aux participants de coopérer et de coordonner le travail du groupe comme au cours d'une réunion synchrone, à l'aide de vues partagées.

Notons que Haake et Haake ne considèrent ici que la coordination implicite à travers l'information partagée.

En général, la coopération d'un groupe de travail est formée de longues périodes de coopération asynchrone (en situation isolée ou séparée) ponctuées de courtes séances de coopération en temps réel (situation faiblement ou fortement couplée).

Nous étendons cette analyse à l'ensemble des activités du groupe. En effet, ces situations de coopération proposée pour les activités instrumentales (modifiant l'espace d'information partagée) existent aussi pour les activités de conversation.

3.2. Favoriser la communication dans le groupe

La communication a longtemps été considérée dans le monde des organisations, dans la communauté informatique et dans le domaine des télécommunications sous l'aspect restrictif de la transmission d'information (d'où l'importance des "Workflows" dans les organisations). Or l'avantage de l'utilisation des outils de communication informatiques est de permettre de séparer l'espace de communication (ou de conversation) de l'espace d'information, tout en favorisant l'interaction entre ces deux espaces. Dans cette optique, le support de la communication de groupe doit faciliter l'utilisation d'un langage à deux niveaux (Double Level Language), en distinguant, comme le dit Robinson [ROBIN 91], ce que l'on dit de ce dont on parle, ce que l'on dit pouvant être l'interprétation culturelle d'un monde de référence plus formel dont on parle. Robinson souligne ses propos en rappelant qu'une conversation (même une conversation à propos d'une conversation) est toujours une conversation à propos de quelque chose. Cette remarque présage de l'ambiguïté de la notion de langage à deux niveaux. Cette ambiguïté explique les difficultés rencontrées par des systèmes tels que The Co-ordinator [FLORE 88][ROBIN 91] malgré la prise en compte du besoin du support de la communication dépassant le stade de l'information. En identifiant l'espace des activités de conversation et l'espace des activités instrumentales comme deux espaces distincts mais interconnectés, les objets des conversations étant les activités instrumentales, ces systèmes peuvent prétendre supporter un langage à deux niveaux. Le problème vient de ce que, dans The Co-ordinator, l'espace de conversation, qui joue le rôle du monde culturel vis à vis de l'espace des activités instrumentales jouant le rôle de monde formel, est lui-même très formel (il est régi par des règles strictes d'utilisation de types de messages structurés) : il nécessiterait un autre niveau de langage pour en parler. La leçon que nous en tirons est qu'un système de support de la conversation doit permettre de structurer formellement l'espace de conversation et de référencer l'espace d'information ou d'activités instrumentales. Il doit aussi permettre de référencer librement les éléments de l'espace de conversation entre eux (en offrant des types libres de messages, d'échanges et de liens).

In general, it can be said that any non-trivial collective activity requires effective communication that allows both ambiguity and clarity. These ideas of ambiguity and clarity can be developed as the "cultural" and "formal" aspects of language as used by participants in projects and organisations. "Computer Support" is valuable insofar as it facilitates the separation and interaction between the "formal" and the "cultural". Applications and restrictions that support one level at the expense of the other tend to fail.

The "formal" level is essential as it provides a common reference point for participants. A sort of "external world" that can be pointed at, and whose behaviour is rule governed and predictable.

The "cultural" level is a different type of "world". It is an interweaving of subjectivities in which the possible and the counterfactual are as significant as the "given" ... interpretation and viewpoint take the place of rules and predictability ...

The formal level is meaningless without interpretation, and the cultural level is vacuous without being grounded"

[ROBIN 91]

Comme le montre la table 1, l'assistance à la communication peut être réalisée par des fonctionnalités différentes selon que la situation de coopération donne lieu à des interactions synchrones⁶ ou asynchrones². Dans les deux cas, il faut donner les moyens d'enrichir à la fois la transmission, la structuration et la situation de la communication.

En temps réel, il s'agit d'offrir des médias contrôlables de communication et des moyens de désignation des objets de la conversation. Les applications sont l'intégration de l'audio- et de la vidéoconférence, les fenêtres partagées (WYSIWIS), la technique du télépointeur.

En temps différé, il s'agit d'offrir des moyens de structuration de la conversation et des moyens d'expression riches. Les messages doivent pouvoir référencer les éléments de l'espace d'information et l'espace de travail qui constituent les objets de la conversation.

⁶Les adjectifs *synchrone* et *asynchrone* s'entendent dans nos travaux dans le sens humain, biologique du temps : deux personnes mènent des actions synchrones (on dit aussi en temps réel) s'ils ont la conscience et la volonté de les réaliser simultanément. Dans le cas contraire, les personnes mènent des actions asynchrones (on dit aussi en temps différé)

CARACTERISTIQUES DE LA COMMUNICATION			
	TRANSMISSION	STRUCTURATION	SITUATION
Synchrone	Audio- ou vidéoconférence. Documents multimédia de la fenêtre publique.	Contrôle de la prise de parole. Médiation par le canal vocal. Contrôle informatique du pont audio ou vidéo.	Fenêtre publique et télépointeur.
Asynchrone	Messages écrits ou vocaux. Attachements de documents multimédia.	Implémentation de la théorie de la conversation pour organiser la communication.	Liens entre les messages, Références à des éléments de l'espace d'objets partagé.

Table 1. Le support de la communication

3.3. Favoriser la cohésion du groupe

Pour renforcer la cohésion du groupe, il faut aider les participants à se connaître. Pour ceci, on peut fournir des informations sociales ou organisationnelles sur les autres participants, qui renseigneront par exemple sur leur statut. Il faut aussi fournir à chaque participant la connaissance de son rôle et du rôle des autres dans le groupe. Il est de plus souhaitable qu'il soit possible de proposer à chaque participant un rôle adapté et les moyens (outils, droits) de le remplir. Il faut donc un système informatique souple, malléable. Des informations globales partagées situées sur la tâche et les activités actuelles du groupe renforceront l'appartenance objective et subjective au groupe. Pour entretenir la cohésion d'un groupe, il faut pouvoir imprimer des rythmes de travail au groupe et il est aussi important de favoriser la prise de décision en commun (voir l'importance du mode démocratique pour la cohésion des groupes dans [ANZIE 82]). Il est aussi nécessaire de permettre de prendre et confier des responsabilités, de s'engager vis-à-vis du groupe.

Les fonctionnalités que nous identifions pour supporter la cohésion du groupe répondent à la quadruple détermination dont résulte le moral d'un groupe et que Anzieu et Martin extraient des travaux de Milton L. Blum :

- La conscience d'être ensemble et de coopérer,
- Le sentiment d'avoir un objectif⁷,
- La possibilité d'observer un progrès dans la marche vers l'objectif,

⁷Certains théoriciens dénoncent la notion d'objectif commun comme étant équivoque et obscure [SCHMI 90]. Pourtant, les psychologues admettent la définition du Dictionnaire la Psychologie qui retient dans la définition de coopération la volonté des acteurs d'aboutir à un résultat qui soit bénéfique à chacun d'eux [DARCH 94].

- Le fait que chaque membre soit responsable de tâches spécifiques significatives qui sont nécessaires à l'accomplissement de l'objectif.

3.4. Favoriser l'implication individuelle

L'utilisateur montre une résistance naturelle au travail en groupe, il faut donc éviter une cassure entre le travail individuel et le travail pour le groupe. De plus, la frontière entre les activités individuelles et les activités menées pour le groupe est imprécise et fluctuante.

Cooperative and individual activities are inextricably interwoven in daily work practice ... Cooperative work is always conducted by individuals, and yet, individual activities are always punctuated and saturated by cooperative work. An activity carried out individually may be - or may any time become - part of a wider, loosely coupled cooperative activity.

[SCHMI 93]

On aura tout intérêt à prendre soin de permettre autant que possible l'intégration dans le travail du groupe des résultats d'un travail individuel, réalisés avec des outils personnels. Un utilisateur appréciera plus encore de pouvoir utiliser ses outils personnels pour participer directement au travail du groupe.

People do much of the work alone, without computers or with various tools on different computer systems, and have developed their own work practices for these situations. In order to get groupware accepted, continuity with existing individual work environments is the key issue because users work in either individual or collaborative modes and frequently move back and forth. Groupware that asks them to abandon their familiar tools, methods, and even computer hardware and software, and to learn a new system, just to gain benefits in communication or coordination, is likely to encounter strong resistance.

[ISHII 91]

Quand l'utilisation d'outils spécifiques à la collaboration est nécessaire, il faut pouvoir adapter ces outils au profil et au rôle de chacun, afin que les participants ne se sentent pas obligés de s'aliéner pour participer au groupe. L'emploi du terme aliéner peut paraître excessif mais il faut bien penser que les outils qu'une personne a l'habitude d'utiliser pour réaliser son travail prennent une valeur qui dépasse son simple rôle instrumental : il y a appropriation, adoption de l'outil comme partie de la culture de l'individu.

Permettre d'articuler le travail individuel et le travail du groupe, c'est aussi permettre d'utiliser les outils de travail et les ressources du groupe pour travailler de façon isolée. Ceci permet de prendre peu à peu possession de l'outil, d'accélérer l'adoption de l'outil par le groupe.

Minimiser les contraintes générées par la participation au travail du groupe par l'ouverture en terme de temps, de lieu et de rythme aidera les participants à concilier leur travail individuel et leur participation au groupe. C'est sans doute l'intérêt majeur du développement de nouvelles conceptions et techniques informatiques mais aussi l'enjeu majeur de leur acceptation.

En effet, des collaborateurs, pour mener à bien une tâche en collaboration, ont l'habitude d'utiliser des situations et des moyens de communication différents, et ce souvent de manière opportuniste. Or l'évolution d'une activité dans des situations différentes risque d'échapper au système informatique. Ainsi une négociation commencée en réunion à l'aide d'un outil d'aide à la décision peut se terminer au cours d'un entretien téléphonique où une décision peut être arrêtée, qui échappera certainement au système informatique. Comme on ne peut (et ne veut) permettre à l'ordinateur d'observer et comprendre toutes les actions des individus, il faut permettre aux personnes de reporter facilement (sans une trop grande surcharge de travail) les actions pertinentes qui pourraient avoir échappé à l'ordinateur, dépositaire de la mémoire du groupe. Le laboratoire des technologies de la coopération de l'université de Milan conçoit actuellement un prototype prenant en compte ces considérations [AGOST 94].

Nous avons vu également qu'une résistance au travail en groupe provient de la peur du participant d'être aliéné et dépossédé de son travail. Pour éviter de nourrir cette crainte, il faut avoir la possibilité de garantir l'intégrité des contributions individuelles au sein du travail du groupe.

3.5. Favoriser le partage des ressources du groupe

Comme nous l'avons mis en évidence, le premier degré de coopération est instauré lorsque plusieurs membres partagent un espace d'objets communs que chacun fait évoluer.

Des utilisateurs pourront partager un tel espace si les outils, les activités ou les objets eux-mêmes prennent en charge la gestion des règles d'accès aux objets de l'espace partagé.

On aura besoin de fournir des moyens d'accès partagé (S.G.B.D. coopératifs, Serveurs hypermédia coopératifs, Répertoire...) avec des services de gestion des

conflits⁸, des services d'identification, de recherche et de localisation des objets partagés du groupe. La gestion des accès doit pouvoir s'adapter aux différents modes de coopération (isolé, partagé, faiblement couplé, fortement couplé).

La structuration de l'espace d'information, en fonction de l'organisation du groupe ou des tâches qu'il se fixe facilitera le partage des ressources.

Les droits et moyens d'accès d'un participant à l'espace partagé devront être déterminés en fonction du rôle du participant dans chaque activité et du point de vue que l'activité a sur l'objet considéré.

3.6. Faciliter l'organisation du groupe

Nous identifions sous cet enjeu les besoins qui apparaissent dans un groupe lorsqu'il devient une organisation. Lorsqu'un groupe devient une organisation, tous ses membres ne participent pas à toutes les activités du groupe. Le groupe s'organise. Il se divise alors en équipes de travail. Les membres de l'organisation participent à certaines équipes de travail selon leur statut dans l'organisation, leurs domaines de compétence ou leurs centres d'intérêt. De sorte que, ne travaillant plus en permanence ensemble, les membres du groupe ne se connaissent plus. Or, quand ils se rencontrent dans une équipe de travail, ils ont besoin d'obtenir des renseignements sur les personnes avec lesquelles ils sont amenés à coopérer. Le système doit donc gérer un répertoire évolué de l'organisation, grâce auquel chaque membre peut connaître et se faire connaître. Le répertoire pourra tenir à jour des informations sociales et organisationnelles, par exemple les coordonnées des membres de l'organisation, leur statut, leurs domaines de compétence, leurs centres d'intérêts et leurs participations dans les différents groupes de travail.

Un système informatique qui veut supporter la collaboration dans une organisation doit permettre la constitution dynamique d'équipes de travail et leur intégration dans l'organisation. Les équipes de travail dans une organisation peuvent durer le temps d'une réunion, ou constituer une équipe qui collaborera durant toute la vie de l'organisation. Permettre de constituer des équipes de travail, c'est permettre à des personnes de se regrouper et de construire des espaces de collaboration dans lesquels elles regrouperont les moyens nécessaires à leur collaboration. Ces espaces de collaboration assurent la persistance du travail du groupe. Un même espace de collaboration pourra être utilisé par l'équipe de travail à de multiples occasions et dans

⁸Pour supporter la coopération, la gestion des conflits peut consister à se prémunir des conflits par les mécanismes classiques de verrouillage et de transaction. Nous verrons que lorsqu'il s'agit d'assister la collaboration, cette éviction des conflits est parfois nuisible. La nécessité d'explicitation des conflits est également présentée dans [CROIS 95].

différents modes de collaboration : réunions en temps réel, forum, édition coopérative faiblement couplée... Quelle que soit l'occasion à laquelle un participant de l'équipe de travail entre dans cet espace de collaboration, il doit y trouver les différentes activités de l'équipe dans l'état auquel lui-même et ses collègues les ont amenées par leurs interventions au fil de toutes les occasions passées. Un espace de collaboration constitue un contexte intermédiaire entre l'activité et l'organisation globale. Il réunit un ensemble d'activités dont les contextes organisationnels se recouvrent ou sont connectés. [KUUTT 93] définit le contexte organisationnel d'une activité de groupe à partir des éléments du modèle de l'activité de [ENGES 87] :

- l'outil utilisé par un participant pour accéder à l'objet de l'activité,
- l'objet manipulé dans le cadre de l'activité,
- les règles qui médiatisent la relation entre le sujet engagé dans l'activité et la communauté qui partage l'objet,
- la communauté qui partage l'objet de l'activité,
- la division du travail qui définit la relation entre la communauté et l'objet de l'activité.

Ainsi, un espace de collaboration peut être créé pour réunir les activités menées par une même communauté et/ou utilisant les mêmes classes d'outil. Un groupe peut décider de réunir des activités dans un même espace de collaboration parce qu'elles sont connectées par leurs objets. En résumé, un espace de collaboration répond au problème de la "*contextualisation*" ("*situationalization*") signalée par [KUUTT 93] : il fournit aux activités l'information organisationnelle pertinente. Un espace de coopération est un type de ce que Peter et Trudy Johnson-Lenz appellent un *Container* [JOHNS 91] :

The function of containers is to hold the energy, life, identity and "presence" of the group. They must be big enough to hold the group in the most creatively alive way, but not too big.

A mesure que l'organisation mûrit, elle développe et adopte une culture de groupe, des pratiques : ce sont des outils adaptés aux activités, ce sont aussi des méthodes, des modes et des outils génériques de collaboration. L'environnement informatique d'aide à la collaboration dans une organisation doit permettre ce développement progressif⁹.

Pour permettre aux équipes de coopérer au départ, il doit proposer des espaces de collaboration prédéfinis. Ces espaces de collaboration intègrent des applications

⁹Cette stratégie de développement de l'organisation est comparable au *bootstrapping* présenté par [ENGEL 92]

coopératives qui supportent les différentes activités du groupe¹⁰. Un premier niveau de flexibilité peut être offert aux participants si les applications coopératives offrent le choix entre différents modes de collaboration dans l'activité.

Par la suite, lorsque les membres de l'organisation se sont familiarisés avec une application et les différents modes de collaboration qu'elle propose, ils doivent pouvoir définir de nouveaux modes de collaboration pour cette activité. Lorsqu'ils sont habitués à configurer les différentes activités d'un espace de collaboration, qu'ils sont capables de comprendre le rôle des différentes activités dans les espaces de collaboration, et leurs interactions, les membres de l'organisation doivent pouvoir agencer par eux-mêmes les activités offertes par le système, pour former de nouveaux types d'espace de collaboration. Enfin, quand ils ont bien intégré les notions d'activité de groupe et d'espace de collaboration, les membres d'une organisation doivent pouvoir dériver de nouvelles classes d'activités et les intégrer dans de nouveaux types d'espace de collaboration.

Ainsi, l'environnement informatique de collaboration devient peu à peu le dépositaire de la mémoire organisationnelle. Cette sauvegarde de la mémoire organisationnelle est très importante car l'organisation est un groupe en perpétuel renouvellement : lorsque certains de ses membres quittent le groupe, les pratiques qu'ils ont développées dans l'organisation survivent à leur départ et peuvent continuer d'évoluer. Comparons la mémoire organisationnelle qui se constitue dans notre environnement de collaboration à celle décrite par [CONKL 92]. Les espaces de collaboration persistants mémorisent les artefacts produits par l'organisation mais aussi leur contexte, leurs fondements et le processus de leur création, c'est à dire la mémoire organisationnelle telle que la conçoit [CONKL 92]. Mais l'environnement de collaboration produit également une culture plus vivante qu'une simple mémoire statique car il s'enrichit, se régénère en permanence en intégrant et en réutilisant de nouvelles classes d'activité et de nouveaux types d'espaces de collaboration : il constitue ainsi un système d'augmentation des capacités de l'organisation, au sens où l'entend [ENGEL 92].

Les espaces de collaboration doivent être situés dans le contexte global de l'organisation, afin de s'articuler les uns par rapport aux autres. Cette intégration est réalisée par l'intermédiaire des tâches de l'organisation prises en charge par chaque

¹⁰Notre approche de l'organisation est semblable à la notion performative adoptée par le projet COMIC [BOWER 93]. Comme eux, nous utilisons l'activité [KUUTI 93] comme unité contextuelle élémentaire dans l'organisation. Ce que nous appelons un espace de collaboration est un contexte englobant plusieurs activités : c'est donc une unité contextuelle intermédiaire entre l'activité et l'organisation globale.

équipe. L'équipe de travail est également située dans l'organisation par le statut dans l'organisation des membres qui y travaillent. Enfin, l'équipe de travail est située dans l'organisation en fonction de la partie de l'espace de production de l'organisation sur laquelle elle agit. En résumé les objectifs qu'il se fixe, les membres de l'organisation qui y participent et la partie de l'espace de production de l'organisation sur laquelle il agit, caractérise un groupe de travail vis-à-vis de l'organisation.

SUPPORTER LA COLLABORATION DANS UNE ORGANISATION, C'EST :

<p style="text-align: center;">1 FAVORISER LA COMMUNICATION</p>
<p style="text-align: center;">2 FACILITER LA COORDINATION</p>
<p style="text-align: center;">3 AUGMENTER LA COHESION DU GROUPE</p>
<p style="text-align: center;">4 FAVORISER L'IMPLICATION INDIVIDUELLE</p>
<p style="text-align: center;">5 FACILITER LA PARTAGE D'INFORMATION</p>
<p style="text-align: center;">6 FACILITER LE DEVELOPPEMENT DE L'ORGANISATION</p>

3.7. Récapitulation de la contribution des fonctionnalités aux enjeux

Comme de nombreux états de l'art ont déjà été proposés récemment [RODDE 91][KARSE 94], nous préférons présenter un positionnement de certaines classes de systèmes ou prototypes existants par rapport à nos enjeux.

Pour évaluer la contribution des classes de collecticiels aux enjeux que nous avons identifiés, nous proposons dans ce chapitre la constitution d'une grille de lecture.

Les systèmes intégrés mis à part, un collecticiel n'a pas pour objectif de constituer à lui seul un environnement de travail coopératif. Par conséquent il n'a pas

l'obligation de répondre à tous les enjeux que nous avons identifiés, ni même de répondre entièrement à un seul de ces enjeux. Le but de notre grille n'est donc pas de permettre de porter un jugement de valeur sur les différents produits étudiés mais de constituer une heuristique pour une nouvelle taxonomie. Elle peut être utilisée pour obtenir une idée a priori du rôle qu'un type de collectif peut jouer dans une organisation, plus fine que celle que l'on peut obtenir par la simple classification espace/temps proposée par Rodden. Cependant, les paramètres que nous utilisons sont moins faciles à trancher que la collocation et la contemporanéité. Surtout, l'existence d'une fonctionnalité ne garantit pas la satisfaction de l'enjeu que nous lui associons. Nous estimons que la fonctionnalité proposée a un impact sur la satisfaction de l'enjeu. Si la fonctionnalité est bien réalisée et bien utilisée, le support de l'enjeu est amélioré, sinon il est dégradé. Pour évaluer cette satisfaction effective, il faut évaluer le système en l'expérimentant dans une situation réelle. Mais l'évaluation des collectifs est une tâche très complexe, comme le souligne la figure 3 extraite de la présentation de Co-Learn par Carmel SMITH lors d'un colloque sur l'évaluation des systèmes informatiques de support de l'apprentissage coopératif [SMITH 94].

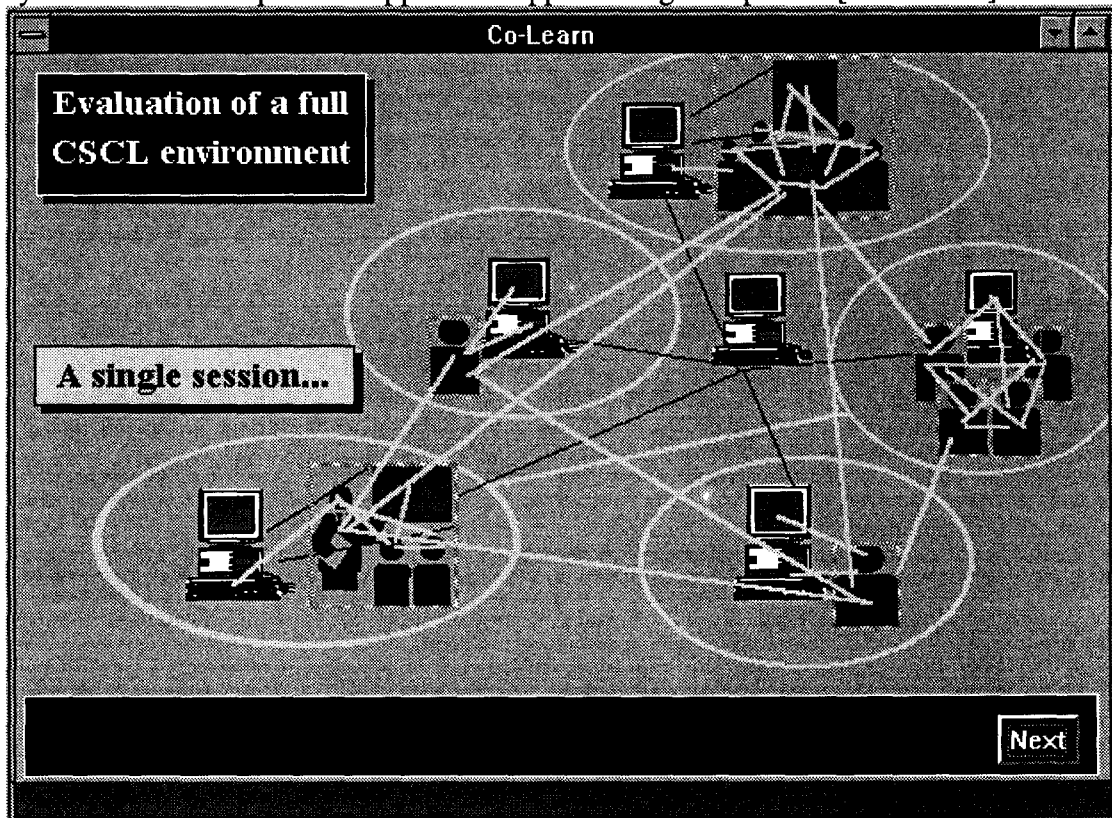


Figure 3. La complexité d'évaluation d'un dispositif informatisé d'apprentissage coopératif [SMITH 94]

Les fonctionnalités que nous répertorions dans la table 2 sont issues de notre analyse pragmatique des enjeux et des articles et documents qui décrivent les systèmes existants.

	FONCTIONNALITES	ENJEUX SUPPORTES
1.	Utiliser différents moyens de communication (vidéo, son, texte)	Communication, <i>Cohésion</i>
2.	Communiquer en temps réel	Communication, <i>Cohésion</i> , <u>Coordination</u>
3.	Communiquer en temps différé	Communication, <u>Coordination</u>
4.	Communiquer en groupe	Communication, <i>Cohésion</i> , <u>Coordination</u>
5.	Mémoriser la communication	Communication, <u>Coordination</u>
6.	Montrer les objets de la conversation	Communication
7.	Désigner les objets de la conversation	Communication
8.	Partager des espaces d'action (= supporter la coopération fortement couplée)	Communication, <i>Cohésion</i> , <u>Coordination</u>
9.	Gérer les risques de conflits d'accès aux espaces d'action communs	<u>Coordination</u>
10.	Structurer la conversation	Communication
11.	Informations sociales et organisationnelles	<i>Cohésion</i> , <u>Coordination</u> , Organisation
12.	Communication informelle	Communication, <i>Cohésion</i>
13.	Attribuer des rôles fonctionnels	<i>Cohésion</i> , <u>Coordination</u>
14.	Connaître les rôles fonctionnels	<i>Cohésion</i> , <u>Coordination</u>
15.	Connaître le contexte des tâches	<i>Cohésion</i> , <u>Coordination</u>
16.	Rythmer le travail du groupe	<i>Cohésion</i>
17.	Prendre des décisions en commun	<i>Cohésion</i> , <u>Coordination</u>
18.	Prendre et obtenir des engagements	<i>Cohésion</i> , <u>Coordination</u>
19.	Connaître l'activité des autres et l'utilisation des ressources dans les différentes situations de coopération	<i>Cohésion</i> , <u>Coordination</u>
20.	Pouvoir faire évoluer les rôles	<u>Coordination</u>
21.	Supporter différentes situations de coopération	<i>Cohésion</i> , <u>Coordination</u> , IMPLICATION
22.	Supporter les transitions entre les situations de coopération	<i>Cohésion</i> , <u>Coordination</u> , IMPLICATION
23.	Réinvestir les résultats du travail individuel dans le travail du groupe	<i>Cohésion</i> , IMPLICATION
24.	Utiliser des outils personnels pour participer au travail du groupe	IMPLICATION
25.	Intégrer des activités non informatisées	IMPLICATION
26.	Adapter les outils à chaque individu	IMPLICATION

27.	Protéger l'intégrité des contributions individuelles au travail du groupe	IMPLICATION
28.	Accéder à une même base d'information	Partage
29.	Adapter aux activités les règles d'accès aux ressources du groupe	Partage
30.	Fournir dans les différentes activités des moyens d'accès à l'espace d'information commun	Partage
31.	Gérer les conflits d'accès à l'espace d'information commun	Partage
32.	Accéder à l'espace d'information commun en situation de coopération isolée ¹¹	Partage
33.	Accéder à l'espace d'information commun en situation de coopération séparée	<u>Coordination</u> , IMPLICATION , Partage
34.	Accéder à l'espace d'information commun en situation de coopération faiblement couplée	<u>Coordination</u> , IMPLICATION , Partage
35.	Accéder à l'espace d'information commun en situation de coopération fortement couplée	<u>Coordination</u> , Partage
36.	Fournir des services de recherche des ressources	Partage
37.	Structurer l'espace d'information commun en fonction de l'organisation	Organisation
38.	Structurer l'espace d'information commun en fonction des tâches	<u>Coordination</u>
39.	Répertorier les membres	Organisation
40.	Gérer la dynamique des rôles organisationnels	Organisation, <u>Coordination</u>
41.	Former des équipes	Organisation
42.	Créer des espaces de coopération persistants	Organisation
43.	Configurer les modes de coopération des activités de groupe	Organisation
44.	Configurer les espaces de coopération	Organisation
45.	Organiser l'univers des espaces de coopération	Organisation
46.	Définir et intégrer de nouvelles activités de groupe	Organisation

Table 2. Répartition des fonctionnalités parmi les enjeux

¹¹Un système inclue cette caractéristique s'il peut garantir à un utilisateur qu'il est effectivement en situation isolée, c'est à dire que personne d'autre ne peut modifier l'information qu'il manipule. Ce n'est pas le cas de la caractéristique 28.

**PARTIE II :
EVALUATION DE LA CONTRIBUTION AUX ENJEUX DES LOGICIELS
D'AIDE A LA COOPERATION**

Dans cette partie, nous analysons plusieurs types de système d'aide à la coopération. Les informations sur ces systèmes ont été obtenues dans les articles qui les ont présentés ou évalués. Seuls quelques uns ont été utilisés au laboratoire Trigone. Par conséquent, il est clair que l'analyse présentée dans ce chapitre ne peut pas être utilisée pour juger de la qualité de ces systèmes.

Nous utilisons la table 2 construite dans la première partie de cette thèse pour mettre en évidence la participation des produits étudiés aux enjeux d'un environnement global de support de la collaboration. Notre but est de montrer l'intérêt que présente l'intégration de différents types de système dans un environnement global.

**TABLE DES MATIERES
DE LA DEUXIEME PARTIE**

CHAPITRE 2 LES LOGICIELS SPECIFIQUES	29
1. SYSTEMES ORIENTES INTERFACE	29
1.1. Les systèmes de partage de fenêtres ou d'écrans.....	29
1.2. Les systèmes de conférence multimédia.....	34
2. SYSTEMES ORIENTES COMMUNICATION	36
2.1. Les systèmes de messagerie électronique améliorés.....	36
2.2. Les systèmes de support de la conversation.....	38
3. SYSTEMES ORIENTES INFORMATION	42
3.1. Les systèmes hypermédia coopératifs	42
3.2. Les systèmes d'édition coopérative	43
CHAPITRE 3 LES SYSTEMES INTEGRATEURS	45
1. Les systèmes exploitant la métaphore de salle virtuelle.....	45
1.1. Milan.....	45
2.2. The Collaborative Desktop	47
2. Les environnements de développement	49
2.1. Pages	49
2.3. ConversationBuilder	51
CHAPITRE 4 SYNTHESE DE L'EVALUATION DE LA SATISFACTION DES ENJEUX	54

CHAPITRE 2

LES LOGICIELS SPECIFIQUES

1. SYSTEMES ORIENTES INTERFACE

1.1. Les systèmes de partage de fenêtres ou d'écrans

1.1.1. Timbuktu

Timbuktu [BREWE 90] permet à un utilisateur (appelé invité) d'observer ou de contrôler un ordinateur Macintosh distant, dont le propriétaire (appelé hôte) a autorisé l'accès requis. Timbuktu a beaucoup évolué entre la première version et la version 3.1 que nous considérons ici. Les nouvelles caractéristiques ajoutées au fil des versions ont amélioré la prise en compte des enjeux de coordination, cohésion et implication et de partage. Plusieurs utilisateurs peuvent observer le même ordinateur, ce qui permet de faire des démonstrations ou des cours à distance. Il n'y a pas de canal de communication intégré : pour communiquer, les utilisateurs doivent être dans la même pièce ou utiliser un dispositif externe de communication (par exemple, une audioconférence). Comme le partage est réalisé au niveau de l'écran, il est possible d'utiliser n'importe quelle application de la machine : les utilisateurs peuvent donc utiliser leurs outils habituels cependant un contrôleur est obligé d'utiliser les outils tels que l'hôte les configure sur sa machine. L'hôte et les invités contrôleurs partagent le contrôle de l'ordinateur. En cas de conflit (plusieurs personnes tentent de déplacer le curseur en même temps), c'est toujours l'hôte qui est prioritaire. L'hôte a la possibilité de modifier dynamiquement les rôles des invités (observateur, contrôleur). Timbuktu permet aussi de transférer des fichiers à destination ou en provenance de l'ordinateur hôte et de visualiser et modifier le contenu du disque distant. Ces fonctionnalités intègrent une forme de coopération en temps différé. Des icônes informent les utilisateurs de l'activité des autres.

Les caractéristiques supportées sont :

- 6 : montrer les objets de la conversation,
- 7 : désigner les objets de la conversation,
- 8 : partager des espaces d'action,
- 9 : gérer les conflits d'accès aux espaces d'action communs (priorité à l'utilisateur hôte),
- 13 : attribuer des rôles,

- 14 : connaître les rôles,
- 19 : connaître l'activité des autres,
- 20 : pouvoir faire évoluer les rôles,
- 21 : supporter différentes situations de coopération,
- 22 : supporter les transitions entre les modes de coopération,
- 23 : réinvestir les résultats du travail individuel,
- 24 : utiliser les outils personnels,
- 26 : adapter les outils à chaque individu,
- 28 : accéder à une même base d'information (partage de disque),
- 39 : répertorier les membres (liste des ordinateurs personnels du réseau).

Ce qui donne le profil :

- 3/10 en communication,
- 5/16 en cohésion,
- 6/21 en coordination,
- 5/9 en implication,
- 1/9 en partage,
- 1/10 en organisation.

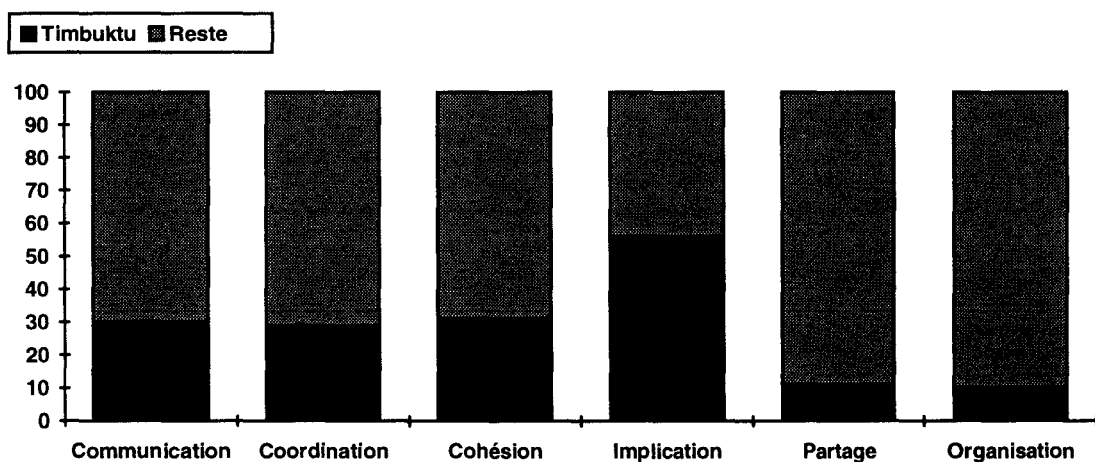


Figure 4. Profil des applications de la classe Timbuktu

1.1.2. Team WorkStation

Team WorkStation [ISHII 91] superpose des sources vidéo pour réaliser l'espace de travail commun. Un seul utilisateur peut manipuler l'information partagée qui sera stockée, les autres ne font que du télépointage.

Les caractéristiques supportées sont :

- 1 : utiliser plusieurs moyens de communication,
- 2 : Communiquer en temps réel,
- 3 : Communiquer en groupe,
- 6 : montrer les objets de la conversation;
- 7 : désigner les objets de la conversation,
- 8 : Partager des espaces d'action,
- 16 : communication informelle,

- 19 : connaître l'activité des autres;
- 21 : supporter différentes situations de coopération,
- 23: réinvestir les résultats du travail individuel dans le travail du groupe;
- 24 : utiliser les outils personnels pour participer au travail du groupe,
- 25 : Intégrer des activités non informatisées,
- 27 : Protéger l'intégrité des contributions individuelles au travail du groupe.

Ce qui donne le profil :

- 7/10 en communication
- 8/16 en cohésion
- 5/21 en coordination
- 5/9 en implication
- 0/9 en partage
- 0/10 en organisation

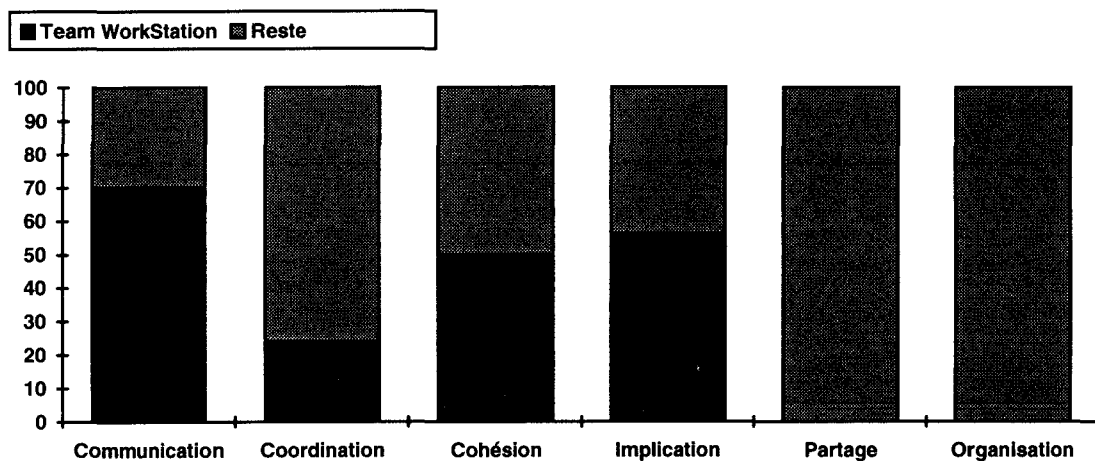


Figure 5. Profil des applications de la classe Team WorkStation

1.1.3. ShX

ShX exploite l'architecture client-serveur du système de fenêtrage XWindow. Le client X est le noyau fonctionnel de l'application et envoie des requêtes d'affichage définies par le protocole X au serveur d'interaction. Le serveur d'interaction analyse et communique au client les actions de l'utilisateur. Le serveur et le client X peuvent s'exécuter sur des machines différentes. ShX ajoute un composant intermédiaire qui gère des communications avec plusieurs serveurs d'affichage pour le compte du même client¹. Les requêtes d'affichage en provenance du client sont multipléxées sur les serveurs et toutes les actions des utilisateurs, captées par les serveurs d'affichage sont communiquées au client partagé. Le composant intermédiaire peut également agir comme un filtre vis-à-vis des actions des utilisateurs, ce qui permet d'allouer le

¹Nous analysons l'architecture logicielle de ce type d'application dans le chapitre 20

contrôle de l'application à un seul client à la fois. ShX comprend un outil qui permet à chaque utilisateur de demander, prendre ou céder le contrôle de l'application. Une fenêtre affiche une liste d'icônes représentant les différents utilisateurs et leur rôle par rapport au contrôle de l'application (demandeur, possesseur, indifférent).

Les caractéristiques supportées sont :

- 6 : montrer les objets de la conversation;
- 7 : désigner les objets de la conversation;
- 8 : partager des espaces d'action;
- 9 : Gérer les risques de conflits d'accès,
- 13 : attribuer des rôles,
- 14 : connaître les rôles,
- 20 : pouvoir faire évoluer les rôles;

- 23 : réinvestir les résultats du travail individuel,
- 24 : utiliser les outils personnels.

Ce qui donne le profil :

- 3/10 en communication
- 4/16 en cohésion
- 5/21 en coordination
- 2/9 en implication
- 0/9 en partage
- 0/10 en organisation

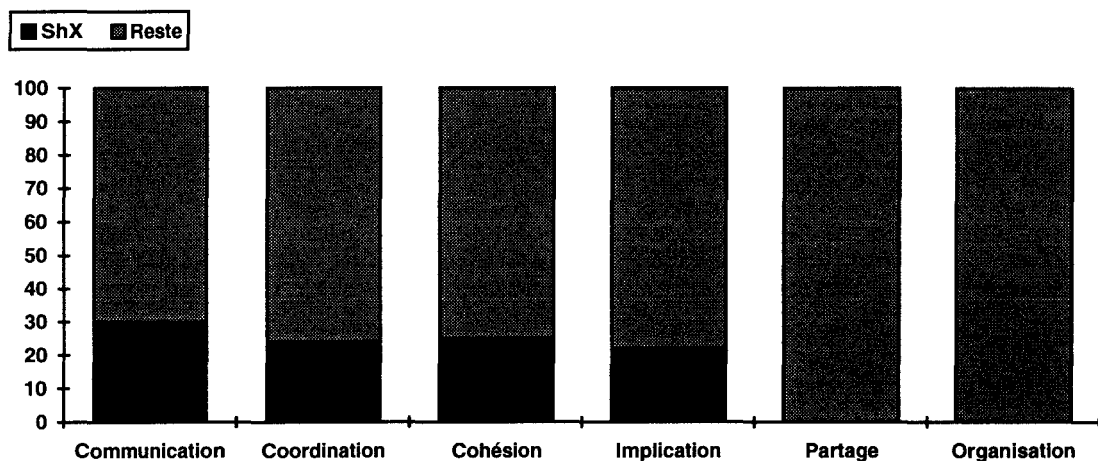


Figure 6. Profil des applications de la classe ShX

1.1.4. SharedX, ShowMe™, Venue™

SharedX [GUST 88], **ShowMe** [SUNSO 92] et **Venue** (Digital) sont des exemples d'applications reposant sur le mécanisme de multiplexage du protocole X que nous expérimentons au laboratoire Trigone. Beaucoup d'applications multi-utilisateurs sur station de travail reposent sur le mécanisme de type ShX. Avec un serveur X sous Windows, il est possible de les utiliser aussi sur ordinateur personnel. Ces applications améliorent ShX en offrant aux utilisateurs des outils complémentaires qui permettent par exemple d'envoyer (et diffuser) des messages textuels aux autres participants et de voter. Il est très difficile d'utiliser ShX et les applications basées sur le mécanisme de ShX à distance sans outil d'audio- ou de

vidéoconférence, à moins que l'application utilisée définisse un protocole strict d'interaction (par exemple une application de jeu de société).

Aux caractéristiques supportées par ShX, il faut ajouter pour ces applications :

- 2 : communiquer en temps réel,
- 4 : communiquer en groupe,
- 12 : communication informelle,
- 17 : prendre des décisions en commun.

Ce qui leur confère le profil suivant :

- 6/10 en communication;
- 8/21 en coordination;
- 8/16 en cohésion;
- 3/9 en implication,
- 0/9 en partage,
- 0/10 en organisation

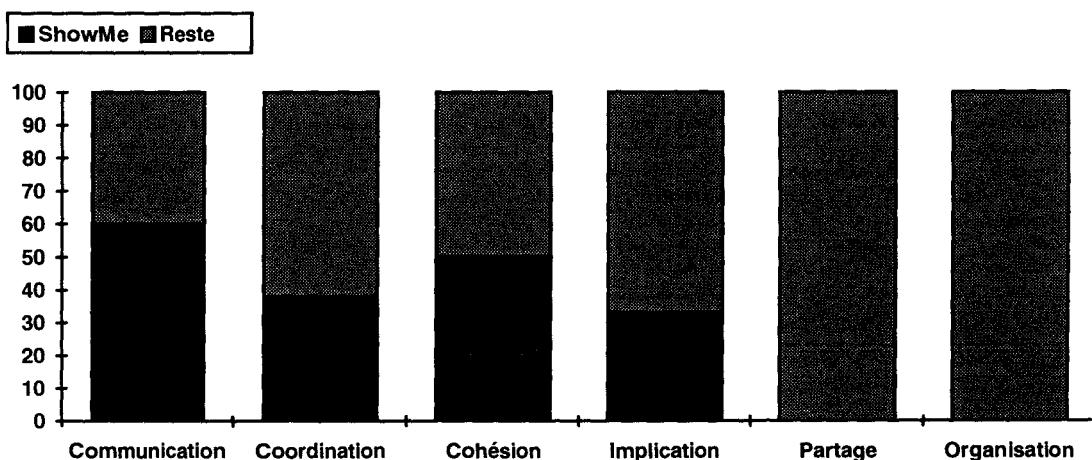


Figure 7. Profil des applications de la classe ShowMe

Ces applications fonctionnent dans l'environnement réseau UNIX. Un partage minimal d'information peut donc être assuré par le partage de système de fichiers, par l'utilisation du logiciel réseau NFS.

En conclusion, les systèmes de partage d'écran ou de fenêtre sont destinés à supporter des coopérations ponctuelles : il n'y a pas de notion de groupe telle que nous l'entendons, donc pas la préoccupation de favoriser la cohésion du groupe. Le système ne considère pas non plus le partage d'information, qui constitue un objectif à plus long terme. Les préoccupations essentielles de la conception de ces systèmes sont l'enjeu de communication, restreint à la présentation d'information, l'enjeu de la coordination, restreint à la coordination à court terme implicite à travers la vue partagée. Ces systèmes répondent assez bien à l'enjeu d'implication parce qu'ils sont

faciles à mettre en oeuvre, parce qu'ils permettent l'utilisation d'applications individuelles ("collaboration transparent" [RODDE 91][BEAUD 92]) et parce qu'ils ne sont utilisés que ponctuellement, et perturbent donc très peu l'organisation du travail des utilisateurs.

1.2. Les systèmes de conférence multimédia

Les systèmes de conférence multimédia offrent à la base les mêmes fonctionnalités que les applications de partage de fenêtre telles que ShowMe. Ils intègrent en plus un canal de communication vocale ou vidéo, pour permettre aux participants de discuter de ce qu'ils font lors d'une utilisation à distance. Ils supportent l'artefact de la réunion, de la conférence, ou de la salle virtuelle. La personne qui ouvre une conférence doit inviter les personnes avec lesquelles elle désire coopérer. Pour rejoindre une réunion ou une conférence, il faut entrer dans la salle virtuelle de réunion appropriée, après y avoir été invité. Vous entrez alors en contact avec l'ensemble des personnes qui sont entrées dans la salle et vous accédez aux écrans communs de différentes applications qui ont été lancées à l'intérieur de la conférence.

[RODDE 91] décrit Rapport, développé par les laboratoires AT&T Bell en 1988, comme exemple de cette classe d'application.

Les caractéristiques supportées, en plus de celles des applications de la classe ShowMe, sont :

1 : Utiliser différents moyens de communication

Ce qui donne le profil suivant :

7/10 en communication,
8/16 en cohésion,
10/21 en coordination
3/9 en implication
0/9 en partage
0/10 en organisation

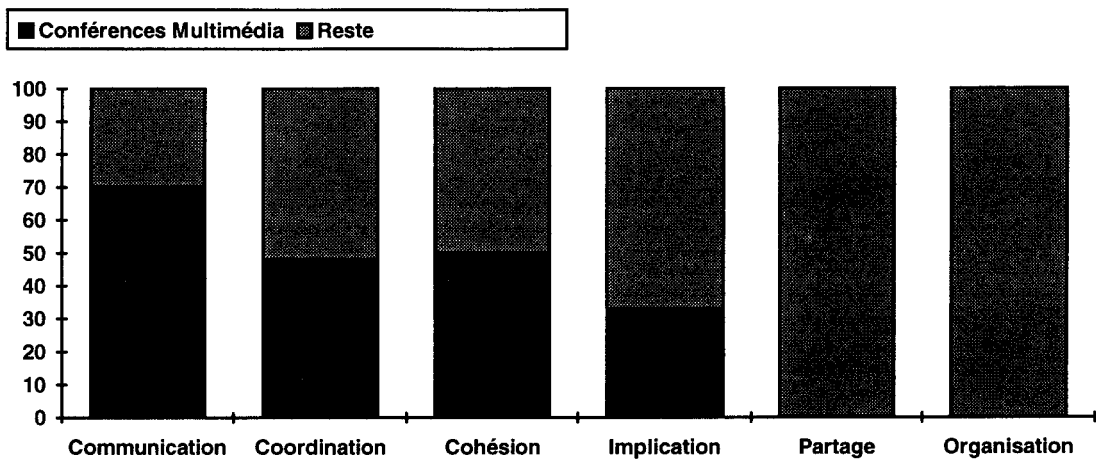


Figure 8. Profil des systèmes de conférence multimédia

2. SYSTEMES ORIENTES COMMUNICATION

2.1. Les systèmes de messagerie électronique améliorés

2.1.1. Information Lens

Information Lens [MALON 87][MALON 92] assiste les utilisateurs en filtrant et triant automatiquement les messages reçus. Il aide également les utilisateurs à créer des messages en proposant des patrons de messages semi-structurés. En plus des agents locaux offrant ces fonctions de filtrage et tri après réception, Information Lens offre des agents qui s'exécutent sur des serveurs de base de messages de groupe : ils sélectionnent pour l'utilisateur les messages dont le destinataire est "Quiconque".

Les fonctionnalités offertes par Information Lens sont :

- 3 : Communiquer en temps différé,
- 4 : Communiquer en groupe,
- 5 : Mémoriser la communication,
- 10 : Structurer la conversation,
- 12 : Communication informelle,
- 28 : Partager une base d'information,
- 30 : Fournir les moyens d'accès adaptés à l'E.I.C.,

36 : Fournir des services de recherche de ressources.

Son profil est donc :

- 5/10 en Communication,
- 3/21 en Coordination,
- 2/16 en Cohésion,
- 0/9 en Implication,
- 3/9 en Partage,
- 0/10 en Organisation

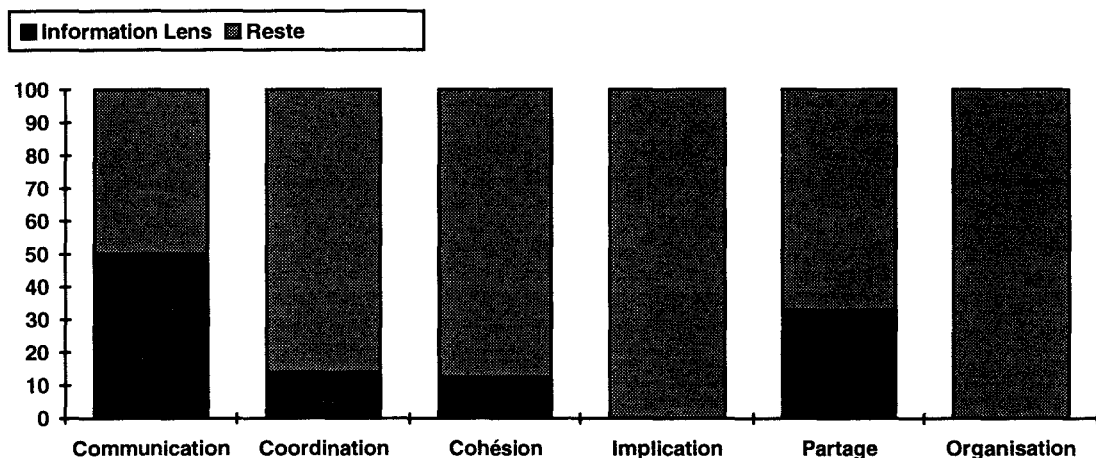


Figure 9. Profil des applications de la classe Information Lens

2.1.2. Andrew Multi-media Message System

Andrew Multi-media Message System [BOREN 88] se veut un système de messagerie intégrant de façon cohérente et efficace les fonctionnalités avancées offertes par les autres systèmes de messageries, l'utilisation des différents média et

l'accès aux divers systèmes de messagerie. Le système offre aussi un service de répertoire. AMS permet également de trier automatiquement les messages que l'utilisateur reçoit. Il offre aussi une fonctionnalité de vote et la possibilité de joindre des données à un message.

Caractéristiques supportées :

1 : Utiliser différents moyens de communication,
 3 : Communiquer en temps différé,
 4 : Communiquer en groupe,
 5 : Mémoriser la communication,
 6 : Montrer les objets de la conversation,
 12 : Communication informelle,
 17 : Prendre des décisions en commun,

23 : Réinvestir les résultats du travail individuel,
 39 : Répertorier les membres.

Le profil est donc le suivant :
 6/10 en Communication,
 4/21 en Coordination,
 5/16 en Cohésion,
 1/9 en Implication,
 0/9 en Partage,
 1/10 en Organisation

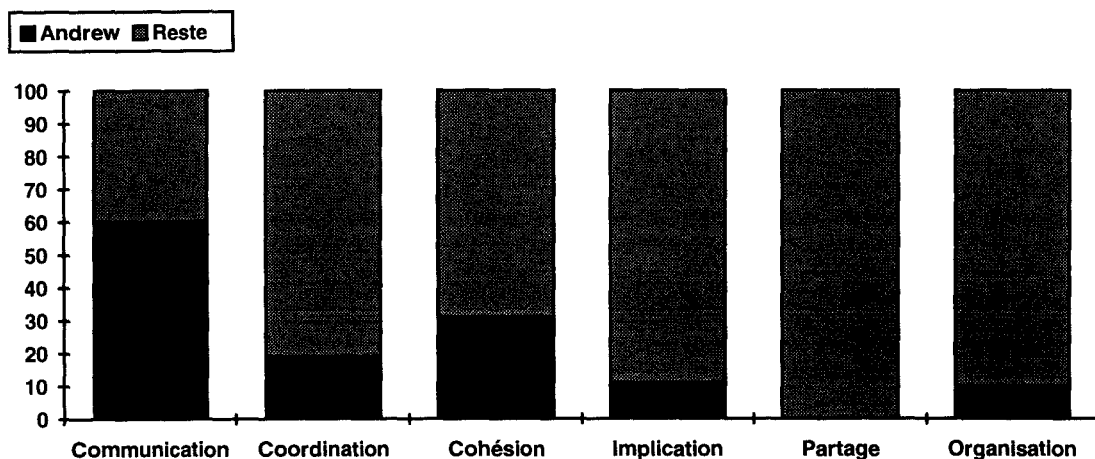


Figure 10. Profil des applications de la classe Andrew Multi-media Message System

2.1.3. SuperCOM

SuperCOM [PALME 91] est un système de conférence qui permet aux utilisateurs d'organiser leurs lectures par conférence, conversation et par toute sous branche d'une conversation. Une conversation est un arbre de messages reliés transitivement par des liens de réponse. Un utilisateur peut jouer le rôle de médiateur pour sélectionner ou classer les messages. Le système permet à l'utilisateur d'utiliser son outil de messagerie habituel. SuperCOM offre aussi des services de répertoire. Ce répertoire peut être utilisé pour ranger des textes. Les concepteurs suggèrent que,

SuperCOM acceptant tout type de contenu de message, y compris du code objet, il puisse être utilisé pour constituer des bases individuelles de logiciels personnels.

Les caractéristiques supportées sont :

1 : Utiliser différents moyens de communication,

3 : Communiquer en temps différé,

5 : Mémoriser la communication,

10 : Structurer la conversation,

12 : Communication informelle,

24 : Utiliser des outils personnels,

28 : Partager une base

d'information,

39 : Répertorier les membres,

41 : Définir des équipes,

42 : Créer des espaces de coopération persistants.

Son profil est donc :

5/10 en Communication,

2/21 en Coordination,

2/16 en Cohésion,

1/9 en Implication,

1/9 en Partage,

3/10 en Organisation.

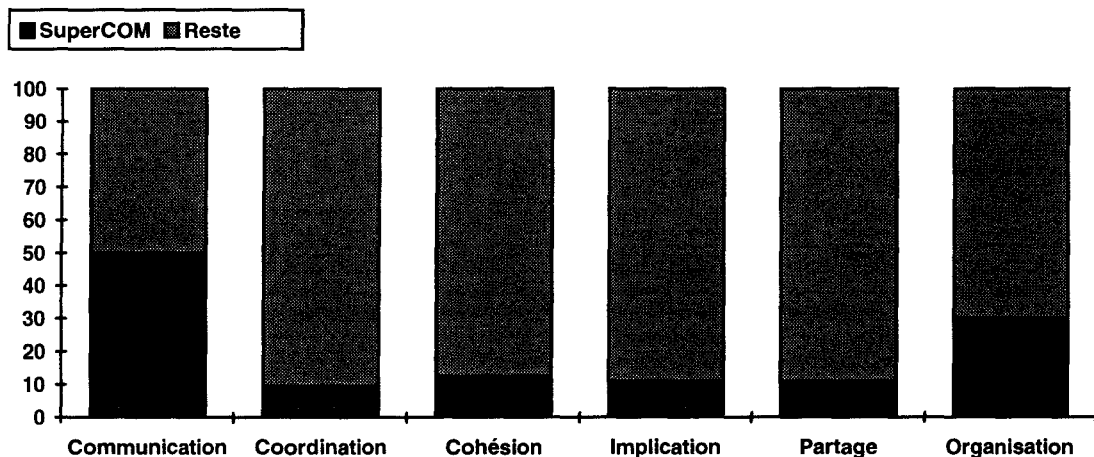


Figure 11. Profil de SuperCOM

2.2. Les systèmes de support de la conversation

2.2.1. The Co-ordinator

The Co-ordinator [FLORE 88][ROBIN 91][RODDE 91][MALON 92], conçu par Flores et Winograd, opérationnalise la théorie de la conversation pour l'action pour assister l'explicitation et la clarification de la négociation pour l'échange d'engagements dans les organisations. Ainsi, Flores et Winograd fondent la conception du système sur l'ontologie du langage action, qu'ils considèrent être au coeur de l'action sociale humaine.

Beaucoup d'organisations ont tenté d'intégrer The Co-ordinator. Certaines intégrations du système furent des réussites. Cependant les communautés de recherche en informatique des organisations et en CSCW tirèrent surtout des leçons de ses

échecs [ROBIN 91]. Flores et Winograd ont eux-mêmes analysé ces réussites et ces échecs dans [FLORE 88].

Ils signalent deux conditions pour la réussite des systèmes basés sur l'ontologie du langage action :

- Le développement dans l'organisation, d'une prise de conscience des problèmes générés par l'ambiguïté des engagements nuancés, et d'une pratique de l'engagement explicite,

- L'intégration du système dans un environnement qui offre d'autres possibilités de communication qui préservent la nuance et l'ambiguïté.

Le but essentiel de The Co-ordinator est de répondre à l'enjeu de coordination. Le système répond cependant à d'autres enjeux mais ce sont soit des effets secondaires, soit des remèdes aux effets pervers du système. Ainsi, The Co-ordinator propose l'utilisation de messages libres pour diminuer la contrainte d'engagement du système, qui est un obstacle à la cohésion du groupe et à l'acceptation individuelle. En effet, demander ou refuser des engagements sans ambiguïté nuit à la cohésion du groupe (c'est souvent ressenti comme un comportement asocial). L'utilisation de messages libres a pour objet de restituer une dimension sociale à la communication en permettant la communication informelle.

Les caractéristiques supportées par The Co-ordinator sont :

- 3 : communiquer en temps différé,
- 5 : mémoriser la communication,
- 10 : structurer la conversation,
- 12 : communication informelle,
- 15 : connaître le contexte des tâches,
- 16 : rythmer le travail du groupe,
- 18 : prendre et obtenir des engagements,
- 19 : connaître l'activité des autres.

Ce qui donne le profil suivant :

- 4/10 en communication,
- 5/16 en cohésion,
- 5/21 en coordination
- 0/9 en implication
- 0/9 en partage
- 0/10 en organisation

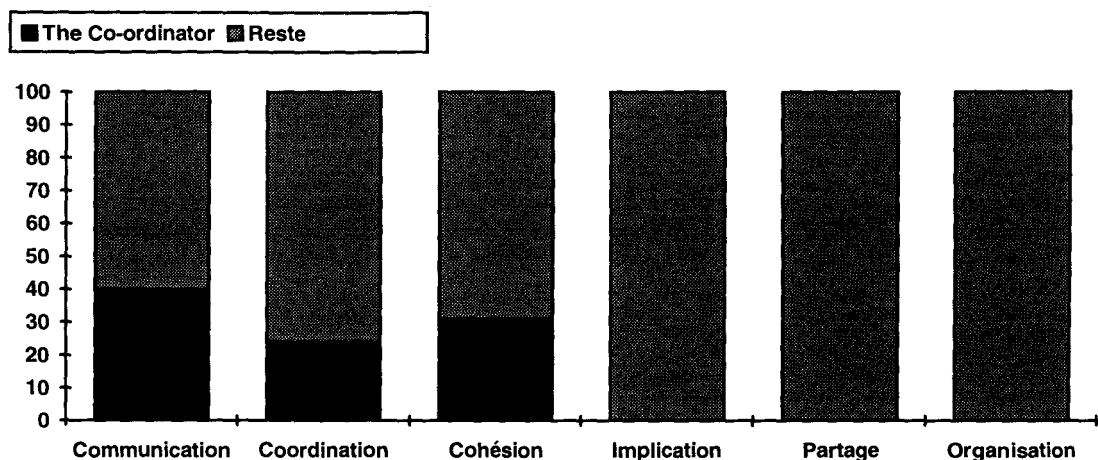


Figure 12. Profil de The Co-ordinator

2.2.2. CHAOS

CHAOS (Commitment Handling Active Office System) [SIMON 93], conçu en 1988 et inspiré de The Co-ordinator et OVAL [MALON 92], se veut être un environnement intégrant communication, gestion d'activité et structuration de l'organisation. La communication permet d'échanger des engagements, engagements à définir et exécuter des activités ou engagements à définir et jouer des rôles structurels. Chaos tient à jour un profil de chaque utilisateur, qui est mis à jour en fonction de ses engagements et ses participations aux activités. Un rôle est un ensemble de protocoles, c'est à dire un ensemble de règles qui gouverne le comportement d'un membre du groupe dans les unités d'organisation et/ou dans des activités.

Les caractéristiques supportées par CHAOS sont :

- 3 : Communiquer en temps différé,
- 5 : Mémoriser la communication,
- 10 : Structurer la conversation,
- 11 : Informations sociales et organisationnelles,
- 13 : Attribuer des rôles,
- 14 : Connaître les rôles,
- 15 : Connaître le contexte des tâches,
- 16 : Rythmer le travail du groupe,
- 17 : Prendre des décisions en commun,
- 18 : Prendre et obtenir des engagements,

- 19 : connaître l'activité des autres et l'utilisation des ressources,
- 20 : Faire évoluer les rôles,
- 40 : Gérer les rôles organisationnels,
- 43 : Configurer les modes de coopération des activités de groupe,
- 46 : Définir et intégrer de nouvelles activités de groupe.

Nous obtenons le profil suivant :

- 3/10 en Communication,
- 11/21 en Coordination,
- 8/16 en Cohésion,
- 0/9 en Implication,
- 0/9 en Partage,
- 4/10 en Organisation

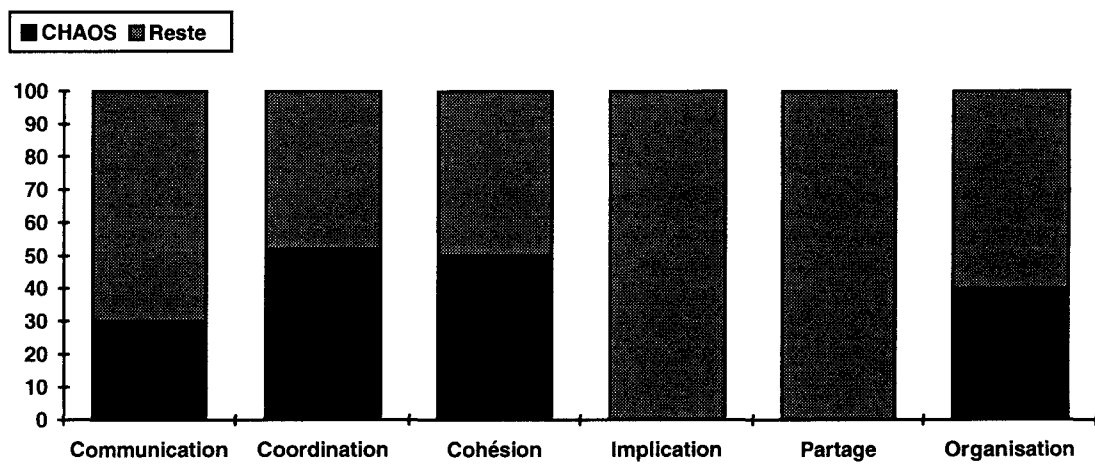


Figure 13. Profil des applications de la classe CHAOS

3. SYSTEMES ORIENTES INFORMATION

3.1. Les systèmes hypermédia coopératifs

3.1.1. Cooperative SEPIA

Cooperative Sepia [HAAKE 92] [HAAKE 93] [STREI 94] est un éditeur/explorateur coopératif d'hypertextes. Il gère quatre espaces d'activité : l'espace de structuration, l'espace de planification, l'espace d'argumentation et l'espace de rédaction. Il assiste les auteurs et les lecteurs dans un spectre important de modes de coopération en temps réel et en temps différé (Cf. p. 12). Le système facilite la transition entre ces modes de coopération. En mode de coopération fortement couplé (en temps réel), les utilisateurs disposent d'une fenêtre partagée (WYSIWIS), d'un télépointeur chacun, d'une connexion sonore et visuelle.

Les caractéristiques supportées sont :

- 1 : utiliser différents moyens de communication,
- 2 : Communiquer en temps réel,
- 3 : Communiquer en groupe,
- 6 : Montrer les objets de la conversation,
- 7 : désigner les objets de la conversation,
- 8 : partager des espaces d'action,
- 9 : gérer les risques de conflits d'accès aux espaces d'action,
- 12 : communication informelle,
- 19 : connaître l'activité des autres dans les différentes situations de coopération,
- 21 : supporter différentes situations de coopération,
- 22 : supporter les transitions entre les situations de coopération,
- 28 : accéder à une même base d'information,
- 30 : fournir dans les différentes activités des moyens d'accès à l'espace d'information commun,
- 31 : gérer les conflits d'accès à l'espace d'information commun,
- 32 : accéder à l'E.I.C. en situation isolée,
- 33 : accéder à l'E.I.C. en situation partagée,
- 34 : accéder à l'E.I.C. en situation faiblement couplée,
- 35 : accéder à l'E.I.C. en situation fortement couplée,
- 36 : fournir des services de recherche des ressources.

Le profil obtenu est le suivant :

7/10 en Communication,

9/21 en Coordination,

8/16 en Cohésion

4/9 en Implication,

8/9 en Partage,

0/10 en Organisation

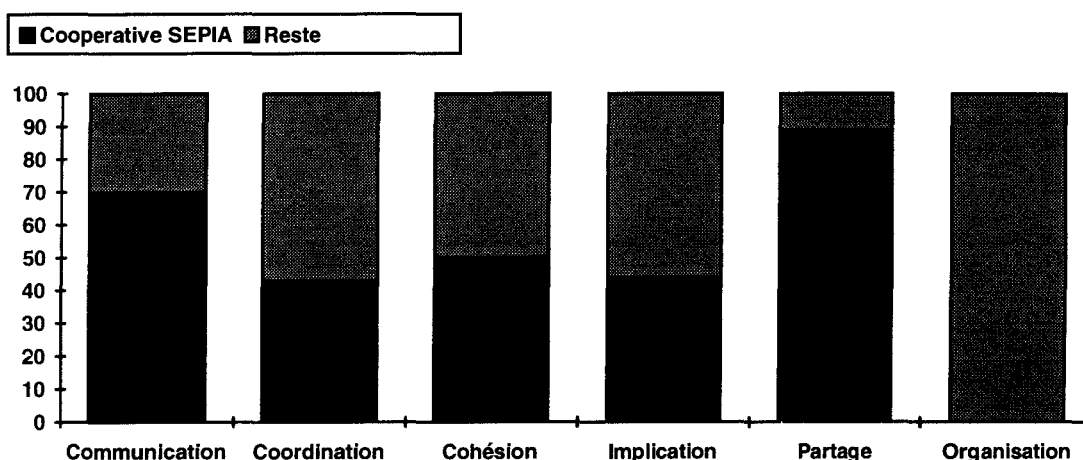


Figure 14. Profil de Cooperative SEPIA

3.2. Les systèmes d'édition coopérative

3.2.1. Alliance

Alliance [DECOU 92] [DECOU 94] est un éditeur coopératif de document structuré. Il permet pour chaque fragment d'attribuer dynamiquement des rôles (gestionnaire, rédacteur, lecteur) qui donnent des droits d'accès différents (créer, détruire des fragments, attribuer des rôles, modifier, lire le contenu d'un fragment). Un seul rédacteur est actif en même temps sur un fragment donné. Alliance supporte donc les situations isolée, séparée et faiblement couplée, mais pas la situation fortement couplée. Les utilisateurs sont tenus au courant des rôles et des actions de leurs collaborateurs par l'insertion d'icônes dans le document. Il est possible d'affecter un rôle à un groupe d'utilisateurs. Chaque utilisateur peut choisir le mode de rétroaction de groupe : aucune rétroaction, rétroaction par notification, rétroaction par mise à jour automatique.

Les caractéristiques supportées sont

:

13 : attribuer des rôles,

14 : connaître les rôles,

19 : connaître l'activité des autres et l'utilisation des ressources,

20 : pouvoir faire évoluer les rôles,

23 : protéger l'intégrité des contributions individuelles au travail du groupe,

28 : accéder à une même base d'information,

29 : adapter aux activités les règles d'accès aux ressources du groupe,
 31 : gérer les conflits d'accès à l'espace d'information commun,
 32 : accéder à l'E.I.C. en situation isolée,
 34 : accéder à l'E.I.C. en situation faiblement couplée,
 38 : structurer l'espace d'information commun en fonction des tâches,

39 : répertorier les membres,
 41 : définir des équipes.

Ce qui confère le profil suivant :
 0/10 en communication,
 6/21 en coordination,
 3/16 en cohésion,
 2/9 en implication,
 5/9 en partage,
 2/10 en organisation

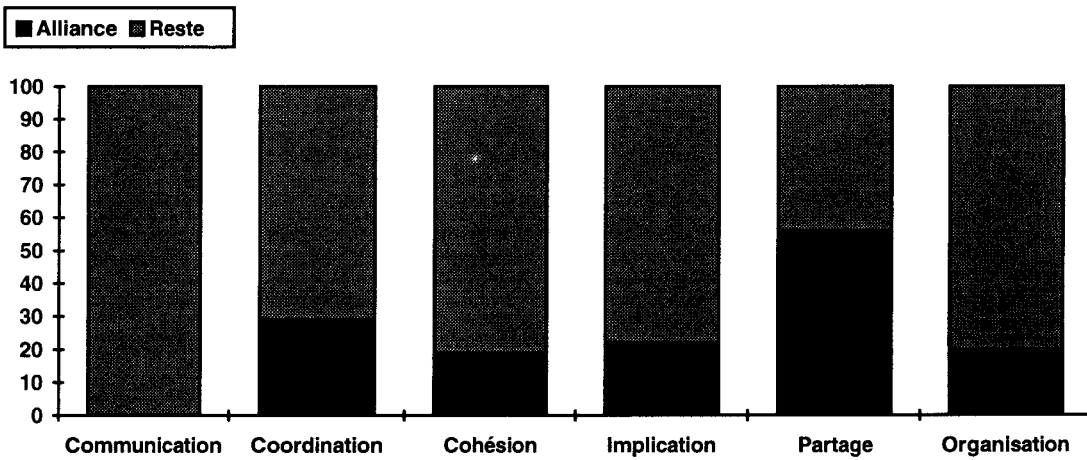


Figure 15. Profil d'Alliance

CHAPITRE 3

LES SYSTEMES INTEGRATEURS

1. LES SYSTEMES EXPLOITANT LA METAPHORE DE SALLE VIRTUELLE

1.1. Milan

Les systèmes qui, comme **Milan** [HAMMA 91], exploitent la métaphore de la salle virtuelle permettent de créer un espace peuplé de salles virtuelles, espace partagé par un groupe de personnes qui peuvent y circuler à leur gré. Une personne pour entrer dans une salle où une réunion est en cours doit obtenir l'accord des personnes qui s'y trouvent. Certaines salles virtuelles sont personnelles : ce sont les bureaux des membres du groupe. Une salle virtuelle de groupe regroupe des participants, des outils et des espaces d'information.



Figure 16. Une salle virtuelle de coopération MILAN [HAMMA 91]

Un écran de projection permet de montrer des documents au groupe, sans les modifier. Le tableau blanc est un espace partagé d'édition de dessins. Les auteurs de Milan ont pu constater que les moyens de communication multimédia (vidéo, voix) offerts par Milan sont essentiels à l'utilisation du tableau blanc. Sur la table, les membres du groupe peuvent poser leurs documents privés que les autres membres sont autorisés à copier. Ceci pourrait être utilisé pour créer des versions différentes d'un même document en copiant le document qui est sur la table, en le modifiant dans son espace privé puis en posant la nouvelle version obtenue sur la table. Notons cependant que les liens de version ne sont pas gérés par le système : les utilisateurs doivent trouver un moyen de faire apparaître ces liens de version par le nom des documents, leur situation sur la table ou par la communication. Chaque membre a un porte-documents qui contient ses documents privés et qu'il emmène dans toutes les salles.

Il est possible de connaître le contenu d'une salle virtuelle, sans y entrer.

Les caractéristiques supportés par de tels systèmes sont :

- 1 : Utiliser différents moyens de communication,
- 2 : Communiquer en temps réel,
- 4 : Communiquer en groupe,
- 6 : Montrer les objets de la conversation,
- 7 : Désigner des objets de la conversation,
- 8 : Partager des espaces d'action,
- 11 : Obtenir des informations sociales et organisationnelles,
- 12 : Communication informelle,
- 15 : Connaître le contexte des tâches,
- 19 : Connaître l'activité des autres et l'utilisation des ressources dans les différentes situations de coopération,
- 23 : Réinvestir les résultats du travail personnel dans le travail du groupe,
- 27 : Protéger l'intégrité des contributions individuelles au travail du groupe,
- 28 : Partager une base d'information,
- 29 : Adapter aux activités les règles d'accès aux ressources du groupe,
- 32 : Accéder à l'E.I.C. en situation fortement couplée,
- 33 : Accéder à l'E.I.C. en situation séparée,
- 37 : Structurer l'espace d'information commun en fonction de l'organisation,
- 39 : Répertorier les membres,
- 42 : Créer des espaces de coopération persistants,
- 45 : Organiser l'espace des espaces de coopération.

Ce qui donne le profil suivant :

7/10 en communication,

8/21 en coordination,

9/16 en cohésion,

3/9 en implication,

4/9 en partage,

5/10 en organisation

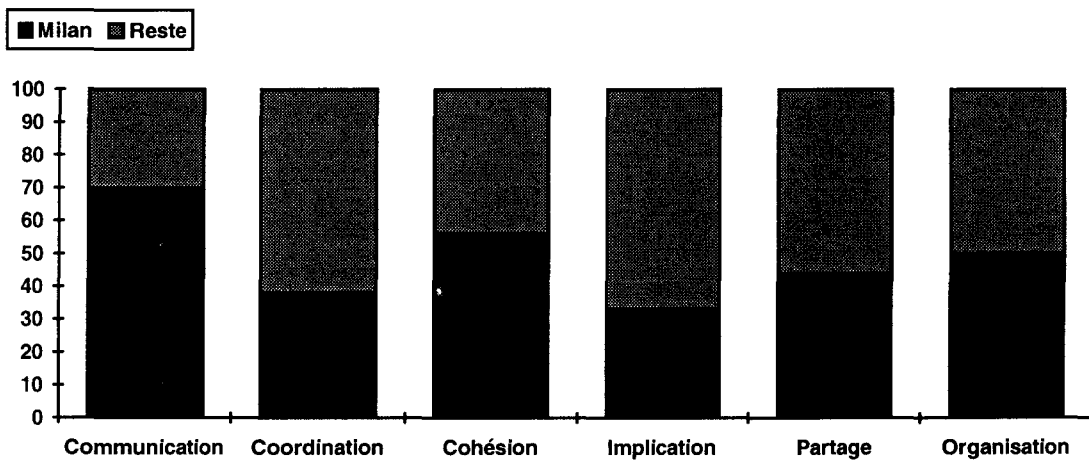


Figure 17. Profil des systèmes de la classe MILAN

2.2. The Collaborative Desktop

Les auteurs de **The Collaborative Desktop** [MARMO 91] pensent qu'il est plus utile de proposer un ensemble d'outils génériques que d'offrir un système intégré qui prescrit implicitement ou explicitement des façons de collaborer. Ils proposent d'offrir parmi les outils une carte d'équipe qui montre une vue générale des activités des participants, un téléphone, un répondeur, un tableau blanc partagé, un outil de collaboration asynchrone et synchrone pour l'édition de documents. Ils proposent également d'offrir une fenêtre vidéo qui permettrait de visionner des documents vidéo stockés, de faire des visioconférences ou de montrer et désigner des objets qui ne sont pas stockés. Ils envisagent également l'utilisation de la métaphore de la pièce. La carte de l'équipe, partagée par tous les membres, montre la salle personnelle de chaque membre sous forme d'une icône et indique quels sont le statut et l'activité du membre ainsi que les outils qu'il utilise. En désignant l'icône d'un participant, on peut obtenir une carte de renseignements sur la personne (photo, coordonnées, projet, emploi du temps). Cette carte peut être utilisée conjointement avec le téléphone, la fenêtre vidéo ou la fenêtre partagée pour appeler un participant. La métaphore de la pièce est aussi utilisée par l'outil de pièce partagée qui permet de créer des pièces où les membres peuvent se rencontrer et disposer de canaux de communication et d'outils partagés pour la coopération. Certaines applications peuvent être définies partageables.

Lorsque deux personnes interagissent en même temps avec une application partageable, elles peuvent se voir et s'entendre.

Les caractéristiques de Collaborative Desktop sont :

- 1 : Utiliser différents moyens de communication,
- 2 : Communiquer en temps réel,
- 3 : Communiquer en temps différé,
- 4 : Communiquer en groupe,
- 6 : Montrer les objets de la conversation,
- 7 : Désigner les objets de la conversation,
- 8 : Partager des espaces d'action,
- 9 : Gérer les risques de conflit aux espaces d'action partagés,
- 11 : Informations sociales et organisationnelles,
- 12 : Communication informelle,
- 15 : Connaître le contexte des tâches,
- 19 : Connaître l'activité des autres,
- 21 : Supporter différentes situations de coopération,
- 22 : Supporter les transitions entre les situations de coopération,

- 24 : accéder à un même espace d'information,
- 32 : Accéder à l'E.I.C. en situation isolée,
- 35 : Accéder à l'E.I.C. en situation fortement couplée,
- 39 : Répertorier les membres,
- 42 : Créer des espaces de coopération persistants,
- 44 : Configurer les espaces de coopération,
- 45 : Organiser l'espace des espaces de coopération.

Ce qui donne le profil suivant :

- 8/10 en communication,
- 11/21 en coordination,
- 10/16 en cohésion,
- 2/9 en implication,
- 3/9 en partage,
- 5/10 en organisation.

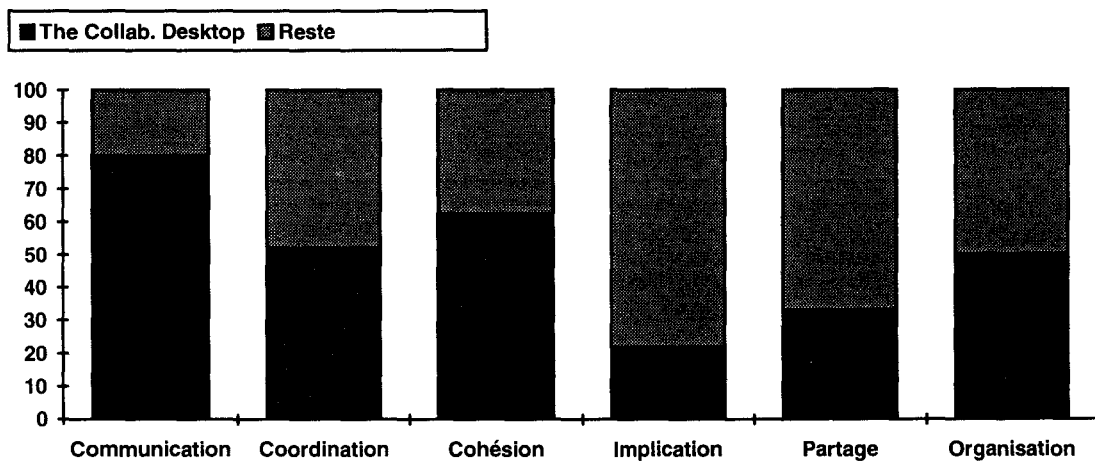


Figure 18. Profil de The Collaborative Desktop

2. LES ENVIRONNEMENTS DE DEVELOPPEMENT

2.1. Pages

Pages [HAMMA 87][MANTY 90][HAMMA 91] utilise des agents utilisateurs semi-automatiques programmables, gérant des bases de formulaires locales, structurées sous forme d'hypertextes. Les utilisateurs et les agents coopèrent en échangeant et en partageant des formulaires multimédia ou des classes de formulaires. Un formulaire est une instance d'une classe de formulaire qui définit des règles de comportement. Un formulaire est composé de champs. Les formulaires et leurs champs réagissent à l'arrivée d'événements. Un champ peut être structuré; sa valeur est interprétée par l'agent utilisateur qui le reçoit. Un champ peut contenir une référence vers un document externe. Le protocole d'échange de formulaires est inclus dans les classes de formulaire. Le protocole d'une application peut être distribué dans les différents types de formulaires de l'application : c'est l'approche recommandée quand la coordination est étroitement liée à l'information échangée. Mais le protocole peut aussi être centralisé dans un formulaire de contrôle qui définit les actions que l'agent doit mettre en oeuvre à la réception des autres types de formulaire de l'application : ce type d'application est basé sur un modèle d'automate à états fini fourni par la théorie du langage action (ce sont les types de formulaires échangés qui déterminent l'évolution de l'activité, et non l'information contenue dans le formulaire). Cette approche de la coopération par échange de formulaires favorise la coopération asynchrone. Pages propose un support de la coopération synchrone par l'édition coopérative de formulaire, grâce à l'utilisation de ShX et d'une communication vocale. Lorsque plusieurs utilisateurs accèdent en même temps au même formulaire, ils entrent automatiquement dans ce mode de coopération synchrone. Mais utiliser un formulaire pour coopérer en temps réel n'est pas naturel : en synchrone, les participants préfèrent utiliser des modes moins formels de coopération. Les utilisateurs peuvent connaître l'activité des autres participants en consultant le plan des agents utilisateurs et en navigant entre les agents utilisateurs. Les bases de formulaires stockent à la fois l'espace de communication et l'espace d'information : certains formulaires constituent des actes de conversation, d'autres sont des documents de l'espace d'information.

Les fonctionnalités que peuvent supporter les applications développées avec Pages sont :

- 1 : Utiliser différents moyens de communication (formulaires multimédia),
- 2 : communiquer en temps réel,
- 3 : communiquer en temps différé,
- 4 : communiquer en groupe,
- 5 : mémoriser la communication,
- 6 : Montrer les objets de la conversation (inclusions de documents dans les formulaires),
- 8 : Partager des espaces d'action (édition coopérative de formulaire),
- 10 : Structurer la conversation (structuration des bases de formulaires),
- 12 : Communication informelle,
- 15 : Connaître le contexte des tâches,
- 16 : Rythmer le travail du groupe (les formulaires comportent un champ date limite et les règles peuvent être activées en fonction du temps),
- 17 : Prendre des décisions en commun (formulaires de négociation),
- 18 : Prendre et obtenir des engagements,
- 19 : Connaître l'activité des autres et l'utilisation des ressources dans les différentes situations de coopération
- 21 : Supporter différentes situations de coopération,
- 22 : Supporter les transitions entre les situations de coopération,
- 23 : Réinvestir les résultats du travail individuel dans le travail du groupe (copier/coller),
- 27 : Protéger l'intégrité des contributions individuelles au travail du groupe,
- 28 : Partager une base d'information (partage des bases de formulaires),
- 30 : Fournir dans les différentes activités des moyens d'accès à l'Espace d'Information Commun,
- 31 : Gérer les conflits d'accès à l'Espace d'Information Commun,
- 32 : accéder à l'E.I.C. en situation isolée,
- 35 : accéder à l'E.I.C. en situation fortement couplée,
- 36 : Fournir des services de recherche des ressources,
- 38 : Structurer l'espace d'information partagé en fonction des tâches.
- 39 : Répertorier les membres,
- 43 : Configurer les modes de coopération des activités de groupe,
- 46 : Définir et intégrer de nouvelles activités de groupe

Ce qui leur confère le profil suivant :

- 9/10 en communication;
- 13/21 en coordination;
- 13/16 en cohésion;
- 4/9 en implication
- 7/9 en partage
- 3/10 en organisation

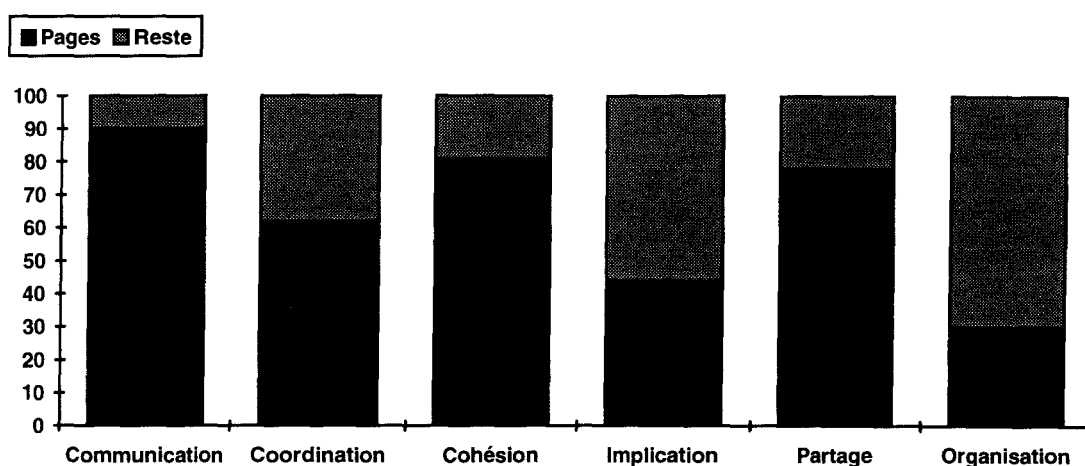


Figure 19. Profil de Pages

2.3. ConversationBuilder

ConversationBuilder [BOGIA 92] est un ensemble d'outils conçus pour fournir une assistance flexible et active aux processus de collaboration. Le système offre des protocoles que les utilisateurs peuvent instancier, définir ou dériver, un hypermédia partagé persistant. Les protocoles définissent les actions offertes à l'utilisateur, quand et à qui ces actions sont autorisées, l'effet de chaque action sur la base d'objets persistante et comment les participants sont avertis des actions. Les auteurs de ConversationBuilder appellent conversation une instance de description de processus de coopération. Il ne s'agit pas de ce que nous avons appelé la conversation, celle qui est explicite entre humains et peut être réalisée par l'échange de messages entre utilisateurs. Il s'agit plutôt de ce que nous appelons un mode de coopération. L'état partagé d'une conversation (*Shared Information State*) décrit quels sont les participants, l'état du processus de coopération et les objets de l'espace d'information partagé qu'il référence. Un état individuel d'une conversation (*Individual Conversation State*) est créé pour chaque participant d'une conversation et décrit les informations telles que les préférences du participant vis à vis de l'activité.

Les caractéristiques supportées sont :

- 1 : Utiliser différents moyens de communication
- 8 : Partager des espaces d'action
- 9 : Gérer les risques de conflits d'accès aux espaces d'action partagés,
- 13 : Attribuer des rôles,
- 20 : Pouvoir faire évoluer les rôles,
- 26 : Adapter les outils à chaque individu,
- 28 : Partager une base d'information,
- 29 : Adapter aux activités des règles d'accès aux ressources du groupe,

- 30 : Fournir dans les différentes activités des moyens d'accès à l'espace d'information partagé,
- 31 : Gérer les conflits d'accès à l'espace d'information partagé,
- 32 : Accéder à l'E.I.C. en situation isolée,
- 34 : accéder à l'E.I.C. en situation faiblement couplée
- 35 : Accéder à l'espace d'information en situation fortement couplée,
- 36 : Fournir des services de recherche des ressources,
- 43 : Configurer les modes de coopération des activités de groupe,
- 46 : Définir et intégrer de nouvelles activités de groupe.

Le profil est alors :

- 2/10 en Communication,
- 6/21 en Coordination,
- 3/16 en Cohésion,
- 2/9 en Implication,
- 8/9 en Partage,
- 2/10 en Organisation

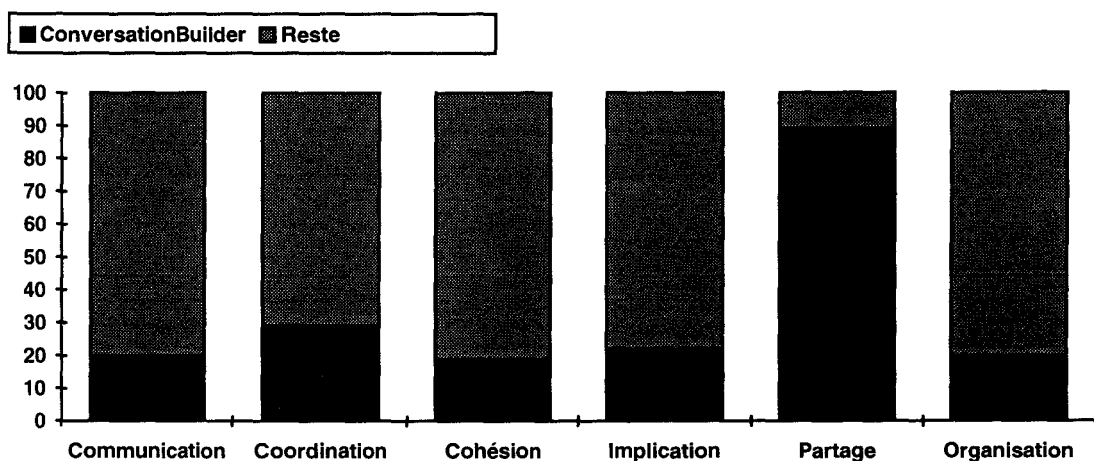


Figure 20. Profil de ConversationBuilder

La carence de certaines des caractéristiques que nous avons identifiées ont été reconnues par les concepteurs du système dans [BOGIA 92]. Ils envisageaient de les intégrer au système.

Ces caractéristiques sont :

- 11 : Informations sociales et organisationnelles et
- 14: Connaître les rôles:

Groups can be used to model organizational structures and participant roles. [...] each participant must be aware of these structures and roles as they dynamically change.

41 : Définir des équipes :

Within CB we have recognized the need for a mechanism like groups for quite some time.

18 : Prendre et obtenir des engagements :

The system should also provide users with a sense of their responsibilities (i.e., actions that others requested that they perform)

42 : Créer des espaces de coopération persistants et

44 : Configurer les espaces de coopération :

Providing an environment allows participants to create local information (i.e. manager roles, work groups, etc.) which may be shared by those conversations running within this environment.

19 : Connaître l'activité des autres et l'utilisation des ressources dans les différentes situations de coopération :

The actions of a participant may affect other participants, possibly in many ways. Participants whose space of actions has changed should have their user interface (UI) updated appropriately.

L'ajout de ces caractéristiques conférerait à ConversationBuilder le profil suivant :

2/10 en Communication,

10/21 en Coordination,

7/16 en Cohésion,

2/9 en Implication,

8/9 en Partage,

6/9 en Organisation

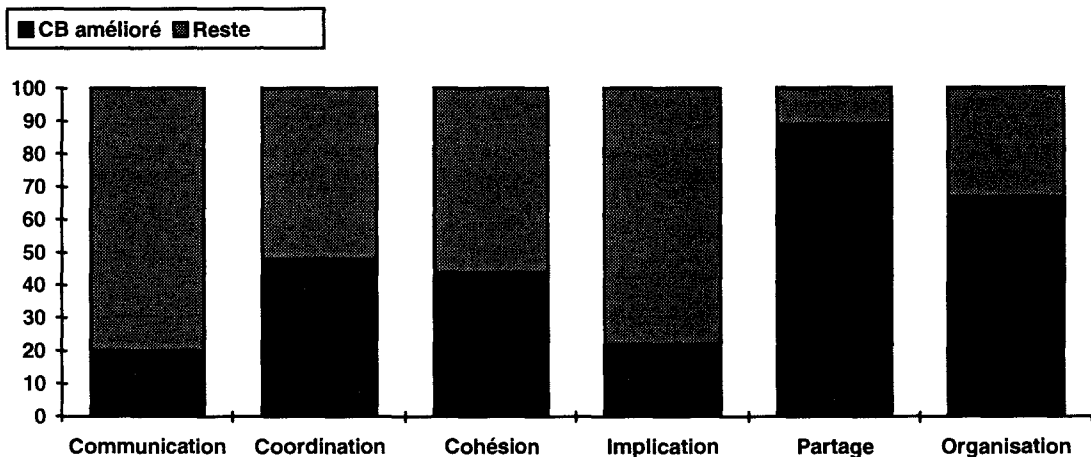


Figure 21. Profil de ConversationBuilder amélioré

CHAPITRE 4
SYNTHESE DE L'EVALUATION DE LA SATISFACTION DES ENJEUX

Système	Comm..	Coord..	Cohésion.	Implication	Partage	Organis..
Timbuktu	30	29	31	56	11	10
Team WorkStation	70	24	50	56	0	0
ShX	30	24	25	22	0	0
ShowMe	60	57	50	33	0	0
Conf. Multimédia	70	48	50	33	0	0
INTERFACE (OPTIMUM)	70	48	50	56	11	10
Information Lens	50	14	12,5	0	33	0
Andrew	60	19	31	11	0	10
SuperCOM	50	10	13	11	11	30
MESSAGERIES (OPTIMUM)	60	19	31	11	33	30
The Co-ordinator	40	24	31	0	0	0
CHAOS	30	52	50	0	0	40
CONVERSATION (OPTIMUM)	40	52	50	0	0	40
Cooperative SEPIA	70	43	50	44	89	0
Alliance	0	29	19	22	56	20
INFORMATION (OPTIMUM)	70	43	50	44	89	20
Milan	70	38	56	33	44	50
Pages	90	62	81	44	78	30
OPTIMUM(Milan, Pages)	90	62	81	44	78	50
The Collab. Desktop	80	52	63	22	33	50
ConversationBuilder	20	29	19	22	89	20
CB amélioré	20	48	44	22	89	67
SYST. INTEGRES (OPTIMUM)	90	62	81	44	89	67

Table 3. Récapitulation de l'évaluation

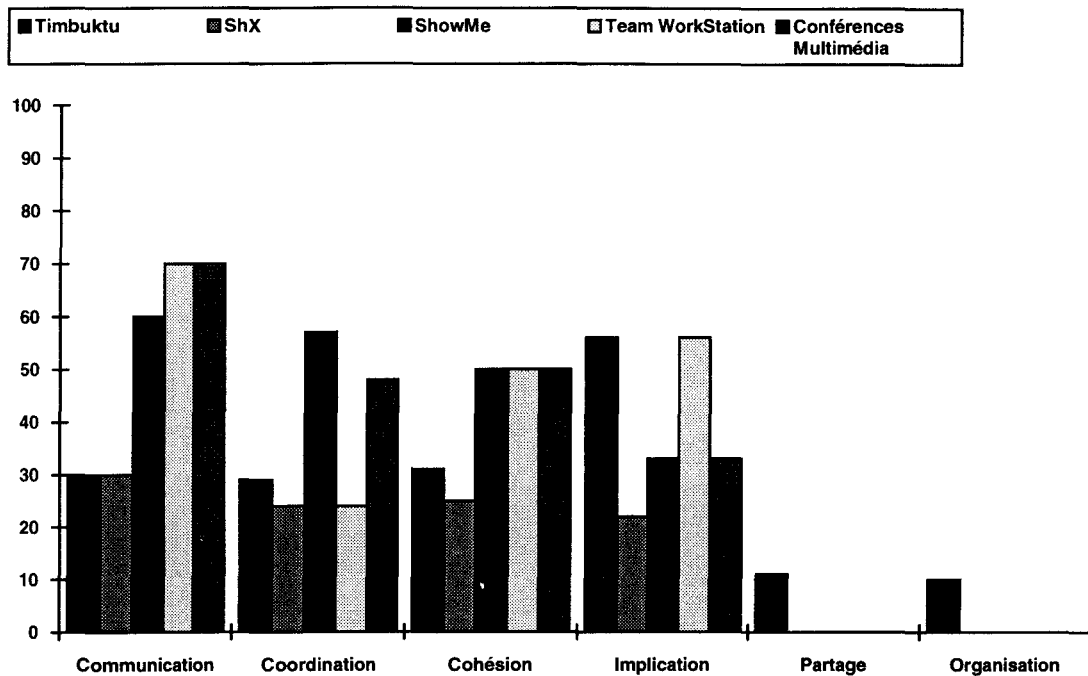


Figure 22. Récapitulation de la participation des collecticiels orientés interface aux enjeux du support de la collaboration

Les collecticiels orientés interface (table 3 et figure 22) supportent les coopérations ponctuelles improvisées, favorisant les échanges (la communication) et la cohésion du groupe. Leurs objectifs atteints de simplicité de mise en oeuvre et d'ouverture témoignent de leur souci d'implication des utilisateurs : c'est un apport important de ce type d'application à un environnement global de coopération. En contrepartie, le rôle essentiel d'un environnement global d'intégration vis à vis de ces applications est d'améliorer leur participation aux enjeux de Partage et d'Organisation, en permettant de sauvegarder pour le groupe le travail réalisé et en situant cette coopération ponctuelle dans le contexte global des activités de l'organisation. L'environnement de coopération doit permettre par exemple de choisir, localiser et avertir l'interlocuteur idéal. La participation des systèmes d'interface de groupe à l'enjeu de coordination se résume à une simple coordination implicite à court terme rendue possible par la visualisation des actions des autres. Un environnement de coopération intégrée doit permettre l'activation de rôles et de modes évolués de gestion de la coordination.

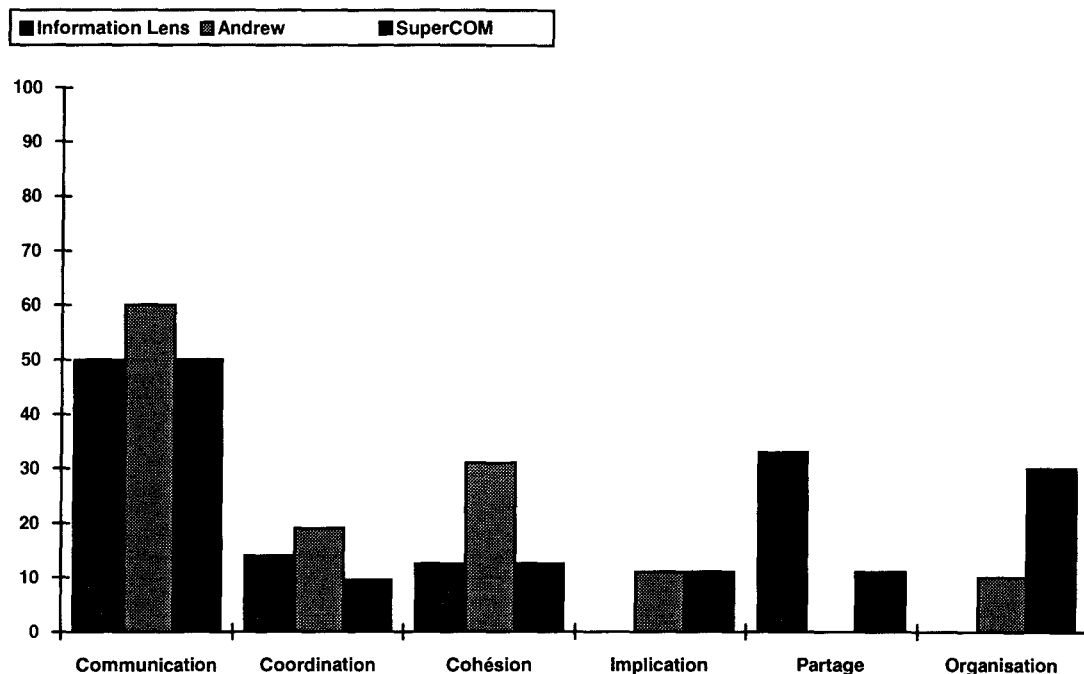


Figure 23. Participation des messageries aux enjeux du support de la collaboration

Les systèmes de messagerie (table 3 et figure 23) apportent à l'environnement des moyens de communication en temps différé, dont les principaux avantages sont le filtrage et la persistance des messages et la structuration de la communication. Un environnement de coopération doit permettre de poursuivre une tâche en utilisant différents moyens de communication, dont les messageries. En effet, pour coopérer, nous utilisons de nombreux moyens de communication dont le choix est souvent une question d'opportunité. Nous pouvons engager une conversation de groupe en temps différé par la messagerie et la poursuivre en temps réel aux cours de réunions. L'intégration dans un environnement de coopération global doit permettre d'utiliser la communication en temps différée pour préparer ou poursuivre un travail mené également au cours de réunions. L'environnement de coopération permet de définir les destinataires des messages en termes d'unités organisationnelles ou d'équipes de travail, de relier la communication à l'espace d'information partagé et de configurer les outils de communication en fonction de l'utilisateur, de l'activité à laquelle il participe et du rôle qu'il y joue.

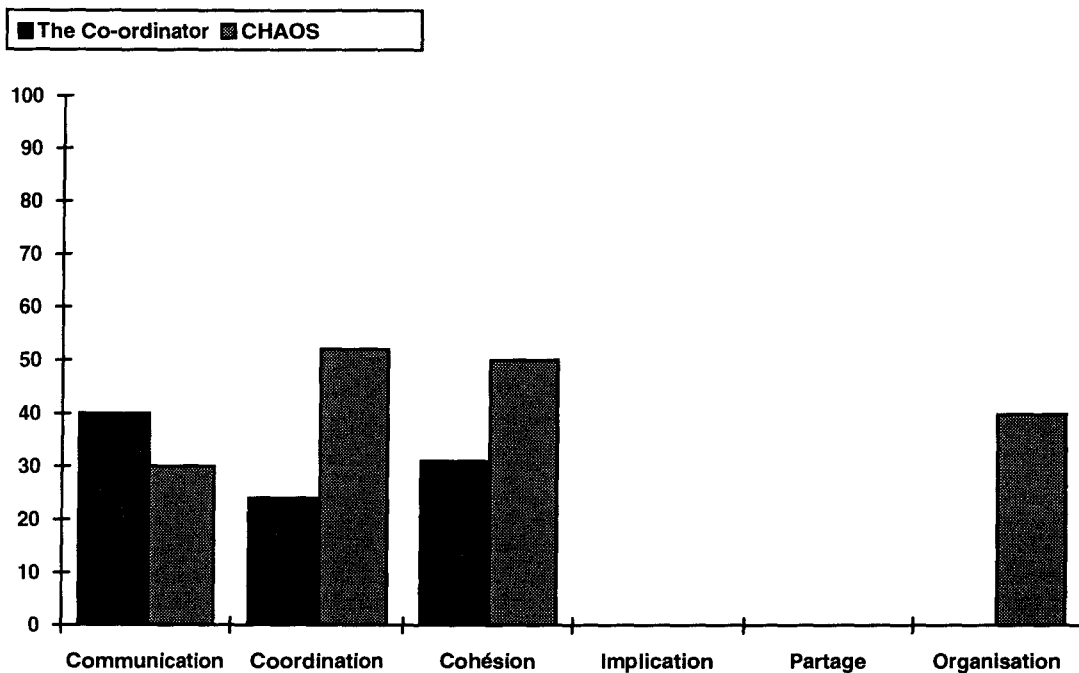


Figure 24. Participation des systèmes de conversation aux enjeux du support de la collaboration

Les systèmes de conversation (table 3 et figure 24) participent essentiellement à l'enjeu de coordination, en permettant aux utilisateurs d'échanger des engagements. En permettant de négocier et définir les rôles structurels, certains de ces systèmes (tels que CHAOS) contribuent à l'enjeu d'organisation. La communication n'est supportée que dans la mesure où elle peut faciliter la coordination. Ces systèmes jouent un rôle important dans l'aspect intégrateur de l'environnement en proposant d'articuler la communication, la gestion des activités et la structuration de l'organisation.

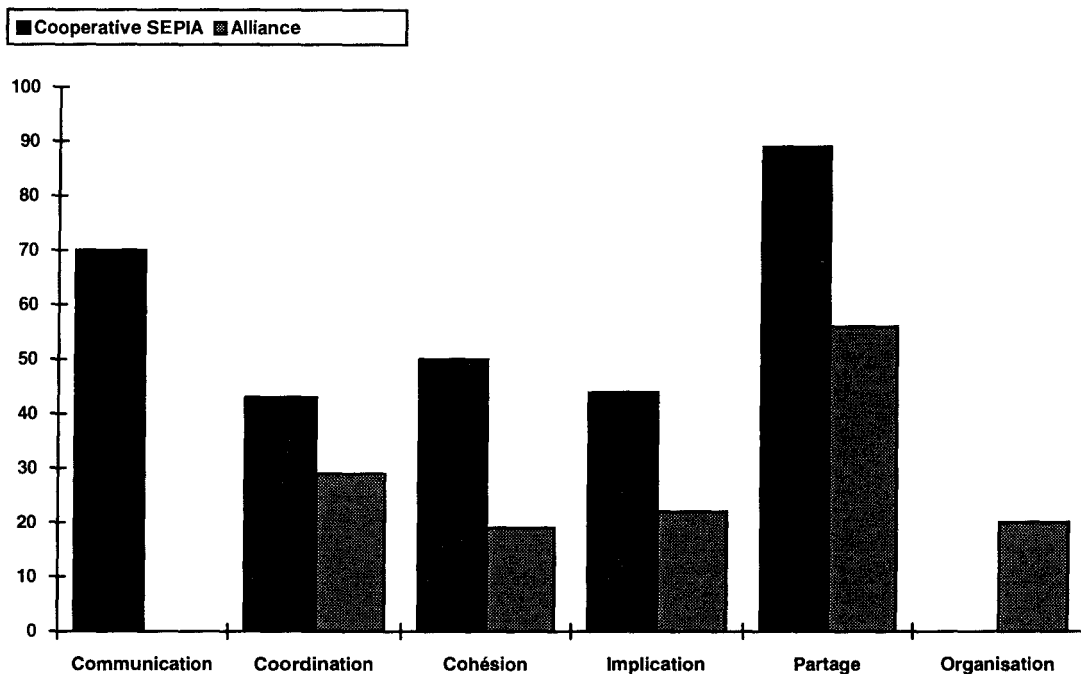


Figure 25. Récapitulation de la contribution des systèmes orientés information au support de la collaboration

La table 3 et la figure 25 montrent que les systèmes orientés information participent d'abord à l'enjeu de partage. Il est intéressant d'observer la différence importante qui existe entre Cooperative Sepia et Alliance. Cooperative Sepia est un système plus récent qui a profité des travaux récents du CSCW. Son profil large montre qu'il est possible de construire un environnement d'aide à la collaboration autour d'un système hypermédia partagé. Un environnement global devrait essentiellement développer les enjeux d'organisation et de coordination, en introduisant les notions de rôle et d'espace de coopération pour situer les activités instrumentales que supportent les systèmes orientés information. L'intégration dans l'environnement global doit permettre de relier les espaces persistants de conversation et d'information, par l'association du système d'information à un système de communication en temps différé (messagerie ou système de conversation).

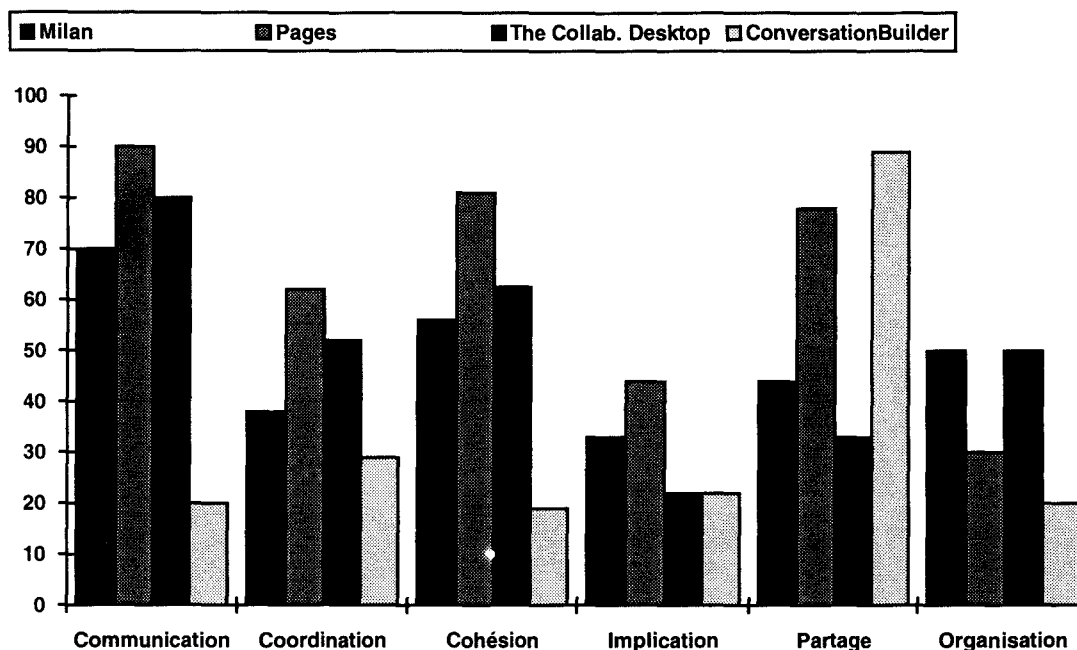


Figure 26. Synthèse de la participation des systèmes intégrés aux enjeux du support de la collaboration

La table 3 et la figure 26 montrent que MILAN et The Collaborative Desktop ont des profils similaires, ce qui reflète la similitude de leur conception articulée autour de la métaphore de Pièce Virtuelle, où l'on trouve des personnes qui mènent des activités de groupe ou individuelles à l'aide d'un ensemble d'outils. Par cette matérialisation des espaces d'environnement que l'on peut créer, configurer, interconnecter, ce sont ces systèmes qui répondent le mieux à l'enjeu d'Organisation. Les systèmes de type Pages jouent un rôle plus important dans la satisfaction des autres enjeux. Notre analyse confirme donc l'intérêt qu'il y aurait à concilier un système basé sur la métaphore de Salle Virtuelle et un système de communication par messages structurés tel que Pages. L'intégration d'un système hypermédia, comme le propose ConversationBuilder, compléterait finalement le système en répondant à l'enjeu de partage.

Conclusion

La satisfaction de l'enjeu d'organisation est l'apport majeur de l'intégration des logiciels de coopération spécifiques dans un environnement de coopération. La Salle Virtuelle est utilisée pour matérialiser et organiser les espaces de coopération. Elle peut délimiter une zone de l'espace global de communication et/ou d'information de l'organisation. Elle permet de réunir ou de donner accès à tous les documents et d'intégrer les multiples moyens de communication ou d'action qu'un groupe de travail peut utiliser pour coopérer sur le long terme.

PARTIE III

LES DISCIPLINES DE RECHERCHE EN INFORMATIQUE FONDATRICES DES ENVIRONNEMENTS GLOBAUX DE COOPERATION

Dans cette partie, qui regroupe les chapitres 9, 10 et 11, nous développons l'analyse de trois aspects de l'informatique. Ils sont importants pour concevoir un environnement distribué de travail coopératif assisté par ordinateur capable de répondre aux six enjeux que nous avons identifiés.

Le chapitre 9 montre comment les bases de données avancées peuvent être utilisées pour concevoir une plate-forme distribuée de support au travail coopératif.

Le chapitre 10 présente ce que doit être le support de la communication humaine pour le travail coopératif.

Le chapitre 11 montre ce que les travaux sur les agents dans plusieurs domaines de l'informatique apportent pour la conception d'un système informatique distribué d'aide au travail coopératif.

TABLE DES MATIERES DE LA TROISIEME PARTIE

**CHAPITRE 5 DU SUPPORT DU PARTAGE D'INFORMATION AU
SUPPORT DE LA COLLABORATION 64**

1. Les systèmes multi-bases	64
2. Les bases de données coopératives	67
2.1. Le choix des bases à objets	67
2.2. Position du problème	68
2.3. Les mécanismes des bases de données avancées	69
2.4. Choix des solutions en fonction des modes de coopération.....	74
2.5. Conclusion	77
3. Base à objets ou système à objets persistants réparti : quel choix pour les premières plates-formes de collaboration ?	78
4. Résumé	81

**CHAPITRE 6 DE LA COMMUNICATION ASSISTEE PAR
ORDINATEUR A LA COLLABORATION INFORMATISEE 82**

1. L'échange privé d'information	82
2. La communication et la répétition individuelles de l'information.....	82
3. Le tri individuel de l'information	83
4. Le filtrage individuel de l'information reçue.	83
5. La diffusion de l'information vers des groupes.	83
6. Le partage de l'information par une communauté.	84
7. Le partage d'une base de messages par un groupe.....	84
8. La circulation de l'information dans une organisation.	85
9. La conversation de groupe : une activité du groupe.....	86

CHAPITRE 7 LA COOPERATION : UNE QUESTION D'AGENTS 89

1. Des agents intelligents ouverts au monde	89
1.1. La résolution de problème distribuée.....	89
1.2. Les systèmes multi-agent.....	90
2. Les agents : des fantômes d'utilisateur dans la machine	91
3. L'agent : un élément clé pour l'adaptation des interfaces homme- machine à la coopération.....	92
4. Les agents communiquent en groupe	93
5. La métaphore d'Agent dans Co-Learn.....	95

CHAPITRE 5

DU SUPPORT DU PARTAGE D'INFORMATION AU SUPPORT DE LA COLLABORATION

1. LES SYSTEMES MULTI-BASES

La première occasion de se heurter au problème de la coopération dans la communauté des bases de données est de prendre conscience de l'intérêt que l'on peut trouver à "fusionner" plusieurs bases de données existantes. Les difficultés que nous identifions alors sont :

- l'hétérogénéité des applications (système d'exploitation, système de communication, S.G.B.D.¹) utilisées par les différents collaborateurs,
- l'hétérogénéité des données manipulées (Schémas des Bases),
- la distribution des applications (S.G.B.D.),
- le maintien de l'autonomie des applications (S.G.B.D.).

Pour répondre à ce besoin, il faut apporter l'interopérabilité entre les S.G.B.D. pour obtenir de façon générale un système multi-base. Selon l'autonomie laissée aux composants, [RONCA 94] distingue :

- les systèmes fortement intégrés : les composants n'ont aucune autonomie, et l'utilisateur ne voit qu'un seul S.G.B.D. : le système multi-base - comme dans le cas d'un S.G.B.D. réparti classique.

- les systèmes fédérés fortement couplés : les composants sont semi-autonomes, ils gardent le contrôle de leur base mais ils coopèrent avec la fédération. Il faut alors mettre en oeuvre entre les composants des mécanismes de communication et de coordination. Une fédération est construite par intégration sélective des systèmes composants, typiquement par la procédure d'intégration de schémas dont le but est de créer un ou plusieurs schémas de données globaux pour la fédération.

- les systèmes fédérés faiblement couplés : les composants ont un haut degré d'autonomie (ils peuvent ignorer l'existence des autres membres) et une intégration très faible (la mise en oeuvre d'opérations sur plusieurs composants est particulièrement difficile). Le maintien de la fédération est de la responsabilité de l'utilisateur auquel la distribution et la duplication éventuelle des données ne sont pas transparentes. En général, les fédérations faiblement couplées n'ont pas de schéma

¹Système de Gestion de Base de Données

global, et le processus typique de mise en oeuvre est l'importation et l'exportation de schémas ou l'utilisation d'un langage multi-base.

Le cas le plus intéressant pour nous est certainement le système fédéré fortement couplé. En effet, c'est le seul cas où des entités semi-autonomes distribuées doivent coopérer, ce qui correspond à la situation d'une coopération humaine informatisée. On peut facilement imaginer que chaque utilisateur humain est représenté dans le système par un ou plusieurs agents² chargés de gérer la communication et la coordination avec les agents représentant des autres utilisateurs humains. L'approche la plus prometteuse de la coopération entre bases composantes est celle où la notion d'objet n'est pas seulement utilisée au niveau des données mais aussi au niveau des bases de données elles-mêmes. Li et McLeod proposent dans [LI 91] une combinaison des technologies utilisées dans les bases fédérées et dans les bases à objets. Plusieurs des concepts propres au modèle objet sont appliqués à l'architecture de la fédération. En particulier, les bases de données, qui peuvent être hétérogènes, sont vues dans la fédération comme des **objets actifs qui coopèrent**, appelés Objectified Component (O.C.). Les O.C. communiquent par **envoi de messages** qui servent aussi bien à **l'échange d'information** qu'à la **négociation** nécessaire au partage d'information.

Pour donner un schéma global à plusieurs bases, il faut réaliser l'intégration des bases. On peut distinguer deux phases : la traduction des schémas et leur intégration.

Lors de la traduction de schémas, l'utilisation d'un modèle canonique permet de résoudre l'hétérogénéité syntaxique et d'enrichir sémantiquement, structurellement et fonctionnellement le schéma local pour faciliter l'intégration de schémas. Pour réaliser ceci, un modèle canonique doit offrir un bon niveau d'expression. Le modèle objet est indiqué dans la littérature comme le mieux adapté.

L'intégration de schémas est généralement menée en cinq étapes : pré-intégration, comparaison, mise en conformité, fusion et restructuration. La richesse du modèle canonique utilisé est très importante dans la phase de fusion : par exemple, la notion de généralisation est fort utile car des concepts similaires peuvent être représentés dans divers schémas avec des niveaux d'abstraction différents.

La gestion de transactions dans un système multi-base reste un sujet de recherche qui nécessite encore beaucoup de travail. Les solutions partielles proposées sont l'utilisation de la sérialisabilité ou de la quasi-sérialisabilité (ordonnancement au niveau global et ordonnancement au niveau local, indépendance des données stockées

²Nous verrons que la métaphore d'agent est utilisée pour l'analyse et la conception d'environnement d'aide au travail coopératif.

sur différents sites et gérées par une même transaction) comme critère de correction. Les S.G.B.D. composants utilisent souvent un protocole de transaction et de coopération à deux phases strict.

Comme le souligne [NICOL 94], il existe de nombreuses analogies entre les problématiques de la fédération des bases de données et le support du travail coopératif :

- articuler le travail individuel et la coopération,
- choisir le niveau de couplage de la coopération,
- faciliter la communication entre les individus,
- pallier les divergences sémantiques.

Nous retenons donc de cette analyse de la fédération des bases de données que **l'approche objet est très prometteuse pour la mise en oeuvre des systèmes multi-bases et sans doute des systèmes coopératifs en général**. D'abord, comme nous l'avons déjà dit, il est reconnu que le modèle objet est très adapté pour la traduction et l'intégration de schémas. D'autre part l'utilisation d'un schéma global objet permettrait d'assurer la transparence de la fédération, grâce en particulier au principe d'encapsulation. On peut ajouter que l'utilisation d'une base d'objets comme intermédiaire pour l'accès à la fédération permettrait, grâce à l'activité de la base, de gérer la fédération de façon transparente. L'approche de Li et McLeod montre l'intérêt de l'approche objet pour la conception des architectures de bases fédérées, mais aussi de systèmes coopératifs en général. Ajoutons que les travaux de l'O.D.M.G. en ce qui concerne la standardisation des S.G.B.D.O.O., leur connexion avec les applications et leur inclusion dans l'architecture CORBA (Common Object Request Broker Architecture) de l'O.M.G (Object Management Group) devraient augmenter encore l'intérêt de l'utilisation de l'approche objet pour répondre aux besoins de coopération entre bases de données et entre applications coopérantes en général.

Pour conclure sur l'approche multi-base, il faut se rappeler qu'elle s'intéresse à la coopération entre bases. Au niveau de l'utilisateur, elle ne facilite la coopération que dans la mesure où elle offre la possibilité de **partager des données entre plusieurs groupes d'utilisateurs**. L'accès à cet espace partagé est toujours envisagé sous une forme **concurrentielle**, par l'utilisation de mécanismes transactionnels qui ont pour conséquence d'**isoler les utilisateurs**. Dans la deuxième approche de la coopération dans le domaine des bases de données, nous nous intéressons aux travaux entrepris pour permettre aux utilisateurs d'accéder de manière coopérative à une même base de données.

2. LES BASES DE DONNEES COOPERATIVES

2.1. Le choix des bases à objets

Les environnements de bases de données traditionnels forcent la sérialisation des transactions, de telle sorte qu'une transaction est totalement isolée des effets d'autres transactions concurrentes. L'utilisation d'une base de données dans le cadre d'applications coopératives nécessite que les actions des différents participants interagissent. Le schéma de contrôle de la concurrence doit être très flexible et les notions de consistance et de correction doivent être définies selon les besoins de l'application. C'est un problème difficile et les systèmes ne font actuellement que fournir à l'utilisateur quelques fonctionnalités pour construire un cadre adapté aux besoins de l'application. Il faut noter tout de suite que les applications nécessitant cet accès coopératif aux bases de données sont des applications hautement interactives, pour lesquelles les bases de données traditionnelles (relationnelles) présentent beaucoup d'inconvénients. Ces applications utilisent donc des bases à objets. Nous ne nous intéressons ici qu'aux fonctionnalités offertes par les bases de données orientées objet pour assister les utilisateurs dans leur coopération. Nous rappelons les avantages des systèmes de gestion de bases d'objets sur les systèmes de gestion de base de données relationnelles, pour le type d'applications qui peuvent être impliqués dans une activité humaine coopérative (les bases de données relationnelles restant tout à fait adaptées aux autres utilisations des bases de données) :

- le modèle de données objet est plus proche de la réalité, grâce à la ressemblance des classes et des objets munis de leur comportement avec les entités du monde réel,

- le modèle de données objet est plus puissant : la représentation des données est flexible. Il permet aux données d'être polymorphes. Les facilités pour l'héritage et l'évolution de schéma confère au système extensibilité, adaptabilité et évolutivité.

- le développement du schéma objet est plus facile : la généralisation donne un schéma mieux structuré, plus concis et capturant mieux la sémantique de l'application,

- le même langage de programmation objet est utilisé pour la manipulation des objets de la base et de l'application : la persistance des objets peut être transparente au développeur,

- le partage d'objet est pris en charge par l'identité d'objet, ce qui garantit l'intégrité référentielle sans besoin pour l'administrateur de la base de se conformer à des règles strictes de normalisation,

- la représentation de la connaissance est unifiée et puissante, elle fournit un cadre cohérent pour intégrer les données et les connaissances (comportements),
- la gestion des transactions et de la concurrence pourrait être plus flexible : les bases de données orientées objets ont l'opportunité de fournir une meilleure gestion de la concurrence, parce qu'elles en savent plus sur les opérations effectuées : ce sont des méthodes déclenchées dans la base en réponse à un envoi de message. La correction pourrait ainsi être garantie sur la base de la sémantique de l'application, plutôt que par l'utilisation de l'approche des conflits en lecture/écriture,
- le langage de requête objet est plus expressif, parce qu'il permet l'utilisation de l'envoi de message et la recherche associative à l'intérieur des requêtes,
- les bases de données objets sont de meilleures candidates pour supporter le travail coopératif.

2.2. Position du problème

Avant d'analyser le rôle des divers mécanismes proposés actuellement par les bases de données pour supporter la coopération, nous identifions quels sont les principes des bases de données traditionnelles et les besoins de la collaboration qui entrent en conflit.

Les contraintes des bases de données traditionnelles sont :

- un utilisateur U1 ne peut voir les modifications apportées par un utilisateur U2 que lorsque U1 et U2 ont enregistré leur transaction (avec succès), d'après le principe d'**isolation**,
- un utilisateur ne doit terminer une transaction que lorsqu'il a réalisé un ensemble d'opérations qui amènent la base dans un état cohérent, d'après le principe de **cohérence**,
- lorsqu'un utilisateur enregistre une transaction, la totalité des modifications qu'il a apportées est enregistrée définitivement dans la base, d'après les principes d'**atomicité** et de **durabilité**,
- si un utilisateur U1 a procédé dans une transaction T1 à des modifications de la base qui entrent en conflit avec les modifications apportées par un autre utilisateur U2 dans une autre transaction enregistrée T2, les modifications proposées dans T1 sont toutes abandonnées
- la base ne communique avec l'application qu'en réponse aux requêtes de l'application, (modèle client actif - serveur passif),

Les besoins de la collaboration qui entrent en conflit avec ces contraintes sont :

- B1 : dans le cadre d'une situation de travail coopératif faiblement ou fortement couplée, un utilisateur B doit être averti des actions menées par A, d'après le principe de rétroaction de groupe,

- B2 : dans une situation fortement couplée, un utilisateur B doit pouvoir intervenir pour négocier les modifications envisagées par un utilisateur A, avant qu'elles ne soient enregistrées définitivement dans la base,

Trois conflits apparaissent :

- C1 : l'incompatibilité de granularité entre la cohérence et la rétroaction de groupe. Par exemple, si la base de données est utilisée par un éditeur de texte coopératif, l'utilisateur pourra juger que les modifications qu'il apporte ne sont cohérentes que lorsqu'il a édité plusieurs paragraphes. Or les autres utilisateurs doivent pouvoir voir les modifications mot par mot, ou lettre par lettre.

- C2 : la passivité de la base par rapport à l'application qui a besoin d'être avertie des modifications apportées dans les autres transactions,

- C3 : la complexité et la longueur des transactions de travail coopératif par rapport au principe d'atomicité. Le type de travail effectué dans une transaction sur une base de données pour le travail coopératif peut être plus complexe et plus long que celui mené au cours d'une transaction traditionnelle. De sorte que l'utilisateur peut vouloir enregistrer à un moment donné un groupe d'objets manipulés dans la transaction et constituant un système cohérent, sans enregistrer les autres objets de la transaction.

Quelles sont les solutions possibles ?

- S1 : diffuser les modifications des objets, sans les enregistrer,
- S2 : avertir les applications des modifications apportées aux objets lors de leur enregistrement,
- S3 : enregistrer des versions différentes d'un même objet,
- S4 : enregistrer une partie réduite des objets manipulés dans une transaction,
- S5 : avertir l'utilisateur de toutes les modifications effectuées sur les objets qui l'intéressent (abonnement ?) depuis sa dernière consultation (historique).

2.3. Les mécanismes des bases de données avancées

2.3.1. La notification

Les mécanismes de notification de mise à jour ont pour rôle d'avertir les autres clients de la base quand un client apporte une modification à un objet persistant. Nous proposons dans le schéma qui suit de mettre en évidence qu'il existe trois niveaux auxquels on peut mettre en oeuvre la notification. Ces niveaux correspondent aux

trois niveaux de manipulation des données lors de l'interaction d'une application avec une base de données³. Le niveau le plus haut est celui de l'application elle-même, le plus bas (stockage) est celui de la base de données partagée, le niveau intermédiaire (transaction) est celui de l'espace de travail privé de l'utilisateur.

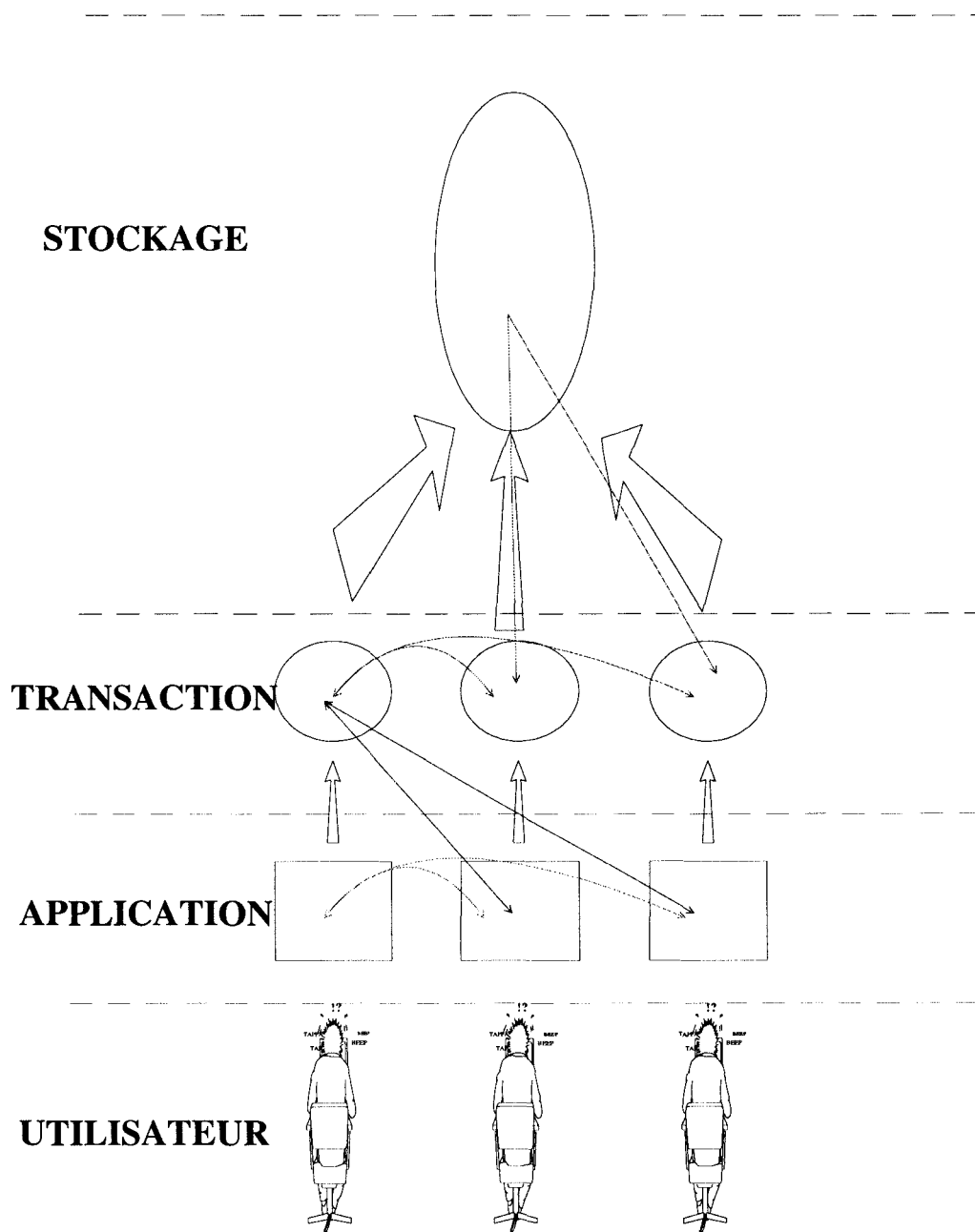


Figure 27. Les trois niveaux de notification

³Ils sont semblables aux couches du modèle de référence des hypertextes de DEXTER [GRONB 94] [HOOGS 90]

L'implantation d'un mécanisme de notification au niveau stockage est étroitement liée au mécanisme transactionnel. Le module de gestion des notifications est averti de la modification d'un objet lorsque l'application qui l'a modifié demande l'enregistrement (commit) de la transaction. Alors que dans un système de base de données traditionnel, il appartient à l'application ou à l'utilisateur de se rendre compte si nécessaire qu'un attribut de l'objet a changé lors de la mise à jour du cache à la fin (par abandon ou enregistrement) de leur propre transaction courante.

Comme [Grønby 94] l'illustre, les notifications telles qu'elles existent dans les systèmes et les prototypes actuels supportant la notification au niveau du stockage peuvent être utiles en situation isolée, séparée ou faiblement couplée mais ne peuvent pas assister la collaboration en situation fortement couplée. En effet, le mécanisme de notification est dans ce cas étroitement lié au mécanisme transactionnel, qui, comme nous l'avons vu, n'est pas adapté à la collaboration fortement couplée. Notons que ceci est vrai pour les deux types de notification au niveau stockage qu'offrent les systèmes actuels : la notification passive et la notification active. La notification passive est d'autant plus inadaptée qu'elle se contente de marquer l'objet modifié et que par conséquent les autres clients ne sont pas notifiés de ce changement tant qu'ils ne consultent pas l'objet. La notification active se contente souvent d'envoyer un courrier électronique aux utilisateurs concernés [AHMED 90, pp. 28-30][ITASCA 90]. Certains systèmes offrant la notification active permettent de redéfinir le comportement actif, défini par une méthode. Mais ce qu'il serait intéressant de définir dans cette méthode est l'envoi d'un message aux représentants de l'objet modifié dans les autres transactions. Or nous n'avons pas connaissance d'un système offrant l'envoi de messages entre transactions : sans doute est-ce pour respecter le principe d'isolation des transactions.

C'est dans la communauté des hypermédia que nous trouvons la mise en oeuvre la plus intéressante des notifications. Dans POSTGRES [ROWE 91], les notifications sont mises en oeuvre au niveau du cache objet. Il semble que la réalisation a été facilitée par l'utilisation sous-jacente d'un S.G.B.D.R.⁴. Il y a ainsi deux niveaux de gestion de l'accès multiple, l'un traditionnel de gestion de la concurrence, l'autre de gestion de la coopération. Vis-à-vis de l'application, la notification est gérée au niveau du cache objet. Du point de vue du S.G.B.D.R. sous-jacent, la notification est gérée au niveau de l'application.

⁴POSTGRES utilisait en 1991 un Système de Gestion de Base de Données Relationnelle mais envisageait de migrer vers une base d'objets pour faciliter l'interaction avec le cache d'objets qu'il gère lui-même.

En fait, POSTGRES propose au niveau application une couche intermédiaire qui aide les applications à gérer la coordination : elle les libère de la tâche d'interaction avec les mécanismes de notification et de verrous qu'elle prend à sa charge pour proposer quatre modes de mise à jour des caches, dont la mise à jour permanente synchrone.

Nous voyons que l'application doit forcément être impliquée dans la gestion de la coopération. Le cas de POSTGRES illustre la nécessité d'un mécanisme de notification réalisé à un niveau supérieur à celui du stockage pour en trouver une exploitation intéressante.

2.3.2. Les versions

La gestion de version est envisagée comme une autre facilité pour assister le travail coopératif. Bien sûr, cette fonctionnalité est peu utile pour supporter le travail coopératif en temps réel fortement couplé. Mais elle peut être intéressante pour supporter le travail coopératif en asynchrone ou en temps réel faiblement couplé, en particulier lorsque la coopération est médiatisée par l'information, c'est à dire quand l'activité du groupe est centrée sur l'édition de documents. Les systèmes hypermédia multi-utilisateurs sont un champ d'application privilégié de cette approche. Cependant la gestion de versions parallèles d'objets dans la base de données ne suffit pas à elle seule à apporter un soutien à la coopération humaine entre utilisateurs. En effet, la notion de version est distribuée sur tous les objets de la base, quels que soient leur niveau d'abstraction et les liens qu'ils ont entre eux.

A elle seule, la gestion de version est sémantiquement trop pauvre et permet juste à plusieurs utilisateurs de la même base de travailler indépendamment sur des versions différentes des mêmes objets.

2.3.3. Les transactions évoluées

La gestion de version est généralement associée à une gestion de transactions imbriquées ou une gestion de transactions longues avec bases privées : ces gestions de transactions continuent, comme les gestions de transactions classiques à isoler les utilisateurs. Il faut donc qu'au-dessus de ces mécanismes de base de gestion des versions soit développé un système d'aide à la coopération orienté tâches, similaire à un système de gestion de projet qui réunit les versions d'objets en fonction de la tâche à laquelle ils sont rattachés et les organise en graphes d'ordonnancement. Take Cover [HAAKE 93] est un exemple de système hypermédia qui propose une aide à la coopération orientée tâches qui *conduit à une gestion des versions de données et à une interface utilisateur étendue permettant à des utilisateurs concurrents de sélectionner*

un certain état de leur travail, d'être informé des changements connexes, et de coopérer avec les autres soit de façon asynchrone soit de façon synchrone (traduit de [HAAKE 93]).

Un tel système d'aide à la coopération doit permettre à tout utilisateur de la base ou à tout participant d'une tâche de voir les versions proposées par les autres. Il doit également supporter différents modes de coopération entre le travail totalement isolé, où chacun développe sa version sans s'occuper des autres, dans le cadre d'une transaction privée, jusqu'au travail coopératif fortement couplé où tous les participants accèdent aux mêmes versions d'objets, afin d'en discuter ensemble, dans le cadre d'une transaction partagée. Dans ce dernier mode de coopération, la coordination des actions des différents participants est totalement liée à la sémantique de l'application. Une rétroaction permanente, un système de communication de groupe garantissant la sérialisation des événements à l'échelle du groupe et un mécanisme d'aide à la résolution des conflits inévitables doivent être mis en oeuvre. Nous pensons que l'on ne peut éviter d'attribuer à l'application un rôle dans la coordination des transactions. Nous partageons d'ailleurs à ce sujet une opinion exprimée par [ANDLE 92] :

The transaction manager can be designed to allow a number of users working concurrently with partially updated objects. As intermediate steps are stored, all users see the effect of the partial changes. Obviously, such a scheme requires a notable amount of programming, as well as application management coordination.

La nécessité de prendre en compte un certain niveau de sémantique pour affiner la gestion des accès simultanés aux objets est illustrée par la mise en oeuvre dans O2 [DEUX 89] de deux modes différents de gestion des accès simultanés. En effet, une gestion plus coopérative des accès au schéma est proposée. Ce qui n'est pas possible en général au niveau des objets, dont la sémantique de modification est liée à l'application, est possible au niveau du schéma parce que la sémantique de modification des éléments du schéma est connu du S.G.B.D..

Plusieurs systèmes de gestion de base de données objets proposent de nouveaux mécanismes avancés de gestion des transactions afin de faciliter la coopération.

Le système de gestion d'hypermédia ITASCA [ITASC 90] et la base de données Versant [VERSA 90], par exemple, proposent la notion de transaction longue (les transactions traditionnelles sont appelées transactions courtes), couplée à la notion de base de groupe ou personnelle et à des mécanismes de "check-in" (qu'on pourrait appeler rapatriement) et "check-out" (qu'on pourrait traduire par importation). Quand un utilisateur importe un objet, il est déplacé de la base partagée vers la base privée de l'utilisateur, de sorte que les autres utilisateurs sont dans l'impossibilité d'accéder à

l'objet, tant que l'utilisateur n'a pas autorisé son rapatriement par un "check-in". Versant propose aussi la notion de transaction imbriquée. Ces mécanismes permettent à chacun de faire évoluer un même système d'une façon différente, indépendamment des autres. Cependant on ne fait que repousser le problème de gestion de la coopération, pour finir par proposer de le résoudre malgré tout par une gestion concurrente des transactions. Ici encore, on ne peut éviter à l'application ou l'utilisateur de prendre en charge les problèmes de coordination.

2.4. Choix des solutions en fonction des modes de coopération

Dans le **mode de coopération isolée**, l'utilisateur est seul à accéder à une partie de l'espace d'information. C'est le mode de travail la plus proche de la transaction traditionnelle. Bien sûr, aucune notification n'est nécessaire. Pour garantir à l'utilisateur cette isolation, l'application peut utiliser le mécanisme classique de **verrous** de la base de données, comme pour une transaction classique. Une autre fonctionnalité de la base de données, qui pourrait être utile aussi dans les transactions classiques, mais s'avère plus utile encore pour supporter le travail de groupe, est la gestion d'un **historique** des modifications des objets de la base. Ainsi, lorsqu'il commence la transaction, l'utilisateur est averti des modifications apportées par les autres utilisateurs depuis sa dernière connexion. Il est possible de limiter les modifications présentées à l'utilisateur par un mécanisme d'abonnement. Ce mécanisme d'historique est très utile par exemple dans les salles de garde ou pour les systèmes d'aide à la conversation en temps différé.

Dans le **mode de coopération séparée**, les utilisateurs travaillent en parallèle sur des **versions** différentes d'un même objet. Les utilisateurs doivent pouvoir être informés de la création d'une nouvelle version ou de l'évolution significative d'une version existante des objets sur lesquels ils travaillent. Les utilisateurs travaillant en mode séparé ne désirent pas être perturbés à la moindre action d'un autre utilisateur : ils ne veulent être informé que des modifications validées. Un mécanisme de **notification au niveau stockage** est tout à fait adapté à ce mode. La granularité de la cohérence égale la granularité de la rétroaction de groupe. Certaines applications de travail de groupe nécessiteront en plus dans ce mode un mécanisme de **transaction longue** avec **base privée** associé aux techniques d'**importation** et **exportation** d'objets. C'est le cas des applications qui donnent lieu à des transactions très longues, au cours desquelles une partie des objets manipulés parvient à un état cohérent. Le mécanisme d'exportation permet à l'utilisateur d'une telle transaction d'enregistrer uniquement cet ensemble cohérent d'objets. Dans ce cas, la notification doit réagir non seulement aux enregistrements des transactions mais aussi à ces exportations. Notons

enfin que le mécanisme d'historique exploité en mode isolé peut aussi être utile pour informer l'utilisateur qui commence une session de travail en mode séparé des modifications apportées depuis sa dernière connexion.

Dans le **mode de coopération faiblement couplé**, les utilisateurs travaillent en même temps sur les mêmes objets. C'est probablement la situation la plus complexe à gérer. C'est par exemple le cas de l'éditeur coopératif de documents structurés Alliance⁵ [DECOU 92] [DECOU 94], qui ne repose d'ailleurs pas sur un système de gestion base de données mais utilise le système de fichiers d'UNIX™. En fait, ce mode de coopération recouvre de nombreuses situations. Les mécanismes utilisés en mode séparé peuvent suffire pour certaines applications. Pour d'autres applications ou dans d'autres contextes, les coauteurs voudront savoir ce que font les autres avant qu'ils ne valident leurs modifications. Si nécessaires, ils pourront ainsi contacter l'autre coauteur pour discuter des modifications en cours avant qu'elles ne soient définitivement enregistrées (et éventuellement utiliser une session de coopération en mode fortement couplé pour travailler ensemble). Dans ce cas, le mécanisme de notification au niveau stockage, qu'il soit associé à l'enregistrement des transactions ou à l'exportation d'objets, n'est plus suffisant : la granularité de la rétroaction est plus fine que celle de la cohérence. On doit alors disposer de la **notification au niveau du cache d'objets ou du noyau fonctionnel de l'application**. La notification peut se déclencher sur les événements de **modification** ou sur les événements de **verrouillage**, selon le détail requis par l'application ou l'utilisateur. On peut également **partager une transaction** ou utiliser des **transactions imbriquées**.

En **mode fortement couplé**, la granularité de la rétroaction est très fine. Elle peut être gérée pour certaines applications (par exemple les applications de vote ou de synchronisation) par la **notification au niveau du cache objet ou du noyau fonctionnel** de l'application. Mais pour d'autres applications ou dans d'autres circonstances ou encore pour certains utilisateurs, elle doit être gérée **au niveau de l'interface** (par exemple pour une application de télépointage ou de fenêtre partagée). Bien sûr, tous les membres d'une session de coopération fortement couplée **partagent la même transaction**. L'utilisation d'une base de données pour supporter la coopération fortement couplée est une nouveauté : jusqu'aujourd'hui, la coopération fortement couplée était entièrement gérée par l'application et l'interface. Dans notre plate-forme, nous utilisons une base à objets active⁶, ce qui nous permet de réaliser les

⁵Ce système a été étudié dans le chapitre 6

⁶Il ne s'agit pas ici d'une base de données active telle qu'on la conçoit habituellement dans la communauté des bases de données, c'est à dire implantant le mécanisme ECA (Événement Condition Action), mais une base telle que GemStone™ ou Objectstore, où les caches sont des espaces d'objets

applications dans des objets persistants de la base : nous n'utilisons pas seulement la base pour stocker des objets-mémoire mais aussi des objets animés d'un comportement. Les versions peuvent être utilisées en mode fortement couplé quand il s'agit de comparer plusieurs versions réalisées au cours de sessions préalables de coopération séparée, pour prendre une décision (choix d'une version ou combinaison de plusieurs versions).

Les transactions longues ou imbriquées peuvent se révéler utiles dans toutes les situations de coopération. En effet, la nature des applications de travail coopératif donne souvent lieu à des transactions longues nécessitant des enregistrements partiels. Parce qu'en mode séparé il s'agit souvent d'applications de cette nature (il s'agit entre autres des applications de conception multi-utilisateur), nous recommandons l'utilisation des transactions longue en coopération séparée.

Il faut finalement remarquer qu'il doit être possible d'être à la fois en mode séparé et en mode isolé ou faiblement ou fortement couplé. En effet, le développement de plusieurs versions d'un même objet peut être confié à autant de sous-groupes. Ainsi un utilisateur travaillant la version d'un objet attribuée au sous-groupe auquel il appartient peut travailler en mode isolé par rapport à ses collègues du sous-groupe et en mode séparé par rapport aux autres sous-groupes. Les différents mécanismes utilisés dans les divers modes de coopération doivent donc être compatibles, c'est à dire pouvoir être utilisés en même temps. Il faut également pouvoir passer facilement d'un mode à l'autre.

La table 4 résume notre analyse.

animés, éventuellement couplés de façon transparente avec les objets d'un espace d'interface (par exemple une interface Gemstone-Smalltalk (GSI), Gemstone C (GCI), ou Gemstone-C++ (GC++I)).

MECANISMES	MODES DE COOPERATION			
	isolée	séparée	faiblement couplée	fortement couplée
Notification au stockage	INUTILE	RECOMMANDE	UTILISABLE	INUTILE
Notification par le cache	INUTILE	INUTILE	RECOMMANDE	UTILISABLE
Notification par l'application	INUTILE	INUTILE	UTILISABLE	RECOMMANDE
Notification par l'interface	INUTILE	INUTILE	INUTILE	RECOMMANDE
Versions	INUTILE	RECOMMANDE	INUTILE	UTILISABLE
Verrous	UTILISABLE	INUTILE	UTILISABLE	UTILISABLE
Files de verrous	INUTILE	INUTILE	UTILISABLE	UTILISABLE
Transaction partagée	INUTILE	INUTILE	INUTILE	RECOMMANDE
Transactions longues ⁷ ou imbriquées	UTILISABLE	RECOMMANDE	UTILISABLE	UTILISABLE
Relation entre les granularités de cohérence (GC) et de rétroaction (GR)	GC < GR	GC = GR	GR >= GC	GR >> GC

Table 4. Choix des mécanismes avancés de bases de données dans les quatre modes de coopération

2.5. Conclusion

Pour conclure, malgré les efforts importants des concepteurs de systèmes de gestion de base de données, la coopération, à la différence de la concurrence, ne peut pas toujours être gérée par la base de données. La politique de gestion de la coopération est liée à la sémantique de l'application. Elle ne peut donc pas être mise en oeuvre uniquement par un système qui gère les conflits d'accès aux objets de façon générique, sans prendre en compte leur sémantique.

Les systèmes de gestion de base de données relationnels ne gèrent que des enregistrements : les types de conflits d'accès et les règles pour les éliminer sont obligatoirement les mêmes pour tous les enregistrements, quel que soit l'objet que l'enregistrement peut représenter.

Dans le cas des systèmes de gestion de base de données objets, la situation est différente car les objets possèdent leur sémantique : chaque objet peut donc avoir la connaissance nécessaire pour gérer les conflits générés par son partage. Cependant, le mécanisme de transaction qui est utilisé pour gérer la persistance des objets (mais aussi garantir l'intégrité de la base et permettre les reprises sur panne) repose sur les principes A.C.I.D.⁸. Ceci implique que chaque transaction accède non pas au même

⁷Nous incluons les mécanismes d'importation/exportation (check-in/check-out) aux transactions longues

⁸Les principes A.C.I.D. garantissent l'Atomicité des opérations (une opération est réalisée entièrement ou elle est abandonnée), la Consistance de la base (une transaction amène la base d'un état consistant à un autre), l'Isolation des transactions (les effets d'une transaction ne sont visibles que dans les transactions qui commencent après son enregistrement), la Durabilité des données enregistrées. [BEVER 93]

objet partagé mais à une version indépendante de l'objet. Ainsi les conflits d'accès ne sont qu'évités par des décisions au niveau du stockage de l'objet. Les objets eux-mêmes ne participent jamais à ces décisions. La résolution du conflit n'a pas pour objet de résoudre le conflit vis-à-vis de l'application ou du groupe mais vis-à-vis du système de stockage. Le conflit est considéré résolu dès l'instant que l'intégrité et la consistance de la base sont assurées. C'est la rigueur des principes des systèmes de gestion de base de données qui constitue l'obstacle majeur à la gestion de la coopération. La relaxation des principes d'Atomicité et d'Isolation permet de résoudre le problème lorsque la granularité de la cohérence est compatible avec la granularité de la rétroaction. Notre conviction est que la gestion de la persistance doit être séparée de la gestion de la coopération et donc de la gestion du multi-utilisateur des transactions, quand la granularité du partage⁹ est inférieure à la granularité de la cohérence. Ces concepts de granularité de la cohérence et du partage nous permettent de nuancer l'affirmation exprimée par Daniels et Cook dans [DANIELS 93].

OBJECT TECHNOLOGY is increasingly being used to construct highly interactive, distributed systems with client-server architectures for use by collaborating workgroups. For these kinds of systems, a conventional database approach is unsuitable. [...]. We then define object sharing and object persistence and show that they are distinct concepts.

3. BASE A OBJETS OU SYSTEME A OBJETS PERSISTANTS REPARTI : QUEL CHOIX POUR LES PREMIERES PLATES-FORMES DE COLLABORATION ?

Nous voyons deux possibilités de réaliser la persistance du système d'information de l'organisation de travail coopératif.

- Première possibilité : on utilise une base de données objet - éventuellement distribuée - pour gérer la persistance. Toutes les stations utilisateurs se connectent à la même base. L'avantage est que l'on continue à bénéficier des services d'un système de gestion de base de données (cohérence, durabilité, requêtes, sécurité). On peut alors appliquer les résultats de l'analyse qui précède pour exploiter la base de données dans les différents modes de coopération. En situation de coopération fortement couplée, le mécanisme de transaction n'est pas utilisé pour supporter le multi-utilisateur mais pour garantir la cohérence et la durabilité des objets stockés par le groupe. La gestion

⁹Nous utilisons dans le même sens les expressions *granularité de la rétroaction* et *granularité du partage*

de la coopération est assurée par les objets eux-mêmes soit dans le cache objet soit dans l'application serveur assurant l'interface avec la base. La coordination entre cette application serveur et le cache objet peut être grandement simplifiée par une gestion transparente de la persistance, telle qu'elle est offerte par les systèmes de gestion de base à objets du type de Gemstone (Cf. Annexe I). Ces bases à objets permettent également de combiner l'exécution de méthodes dans le cache objet et dans l'application d'interface : c'est au concepteur d'application que revient le choix de la stratégie de répartition de l'exécution la mieux adaptée.

Il faut noter que dans ce cas, il est nécessaire que les objets partagés soient des objets complets. Nous entendons par ceci que les objets partagés doivent être au moins réactifs, au sens où ils sont capables de répondre à l'envoi de message et de déterminer leur réaction (l'objet a une structure et un comportement). De plus, il faut pouvoir gérer les flux de messages parallèles provenant des différents utilisateurs à destination des mêmes objets partagés, assurer l'atomicité des diffusions et la sérialisation éventuelle des messages.

Cette gestion de l'activité peut être assurée par l'utilisation d'objets actifs, c'est à dire des objets capables de gérer des files de messages en réception et en émission et de coordonner le traitement de ces messages. Cependant, il faut noter que l'utilisation d'objets actifs entraîne un effort très important de conception et de programmation pour assurer la synchronisation des flux d'exécution. De plus, dans le cadre d'un système de gestion de base de données, l'activité des objets complique la gestion des enregistrements de la transaction. En effet, il est très difficile de détecter un état stable du système d'objets global quand n'importe quel objet est susceptible à tout moment d'exécuter une méthode. Or il est nécessaire de détecter ces états stables pour enregistrer la transaction.

La gestion de l'activité peut être réalisée plus simplement en confiant à certains objets la responsabilité des flux d'exécution (matérialisés par des processus) et donc de la détection des états stables propices à un enregistrement de la transaction.

- Deuxième possibilité : on utilise un système d'exploitation distribué à objets persistants. L'avantage sur la première solution est que les objets sont maîtres non seulement de leur partage mais également de leur persistance. Chaque objet est capable de s'enregistrer sur disque indépendamment des autres objets. Il faut cependant penser que beaucoup d'objets sont dépendants les uns des autres et qu'ils doivent par conséquent utiliser leur capacité à communiquer pour se coordonner afin de garantir la cohérence de l'état global du système d'objets persistants. Actuellement, les systèmes d'exploitation distribués à objets persistants font l'objet d'une recherche

spécialisée. La définition de l'architecture CORBA, en cours d'élaboration par l'O.M.G (Object Management Group), est un grand pas dans le sens du développement des systèmes à objets distribués. L'adoption de la programmation orientée objet implique l'adoption du paradigme de la coopération. Les objets sont conçus comme des **entités autonomes qui coopèrent par l'envoi de messages**. La modularité extrême qu'apporte l'approche orientée objet est d'ailleurs dans ce sens très adaptée à la conception de systèmes répartis ou distribués [WEGNE 92]. C'est pourquoi l'O.S.F.¹⁰ à travers D.C.E.¹¹ [BEVER 93], l'I.S.O.¹² à travers O.D.P.¹³ [LININ 91] et bien sûr l'O.M.G.¹⁴ avec l'O.M.A.¹⁵ et CORBA¹⁶ participent à la promotion de l'approche orientée objet pour la conception des systèmes distribués. D.C.E. utilise encore une approche traditionnelle de la communication entre les objets, c'est à dire le modèle client/serveur, supporté par les R.P.C.. Mais les modèles O.D.P. et O.M.A prévoient et préconisent une évolution vers un **modèle coopératif de la communication**. [WEI 93] explique en effet que les R.P.C. constituent le mécanisme de communication de la programmation distribuée client/serveur alors que **l'envoi de message est le mécanisme de communication de la programmation distribuée coopérative**.

Malheureusement, les systèmes à objets distribués tels que l'architecture CORBA ne sont actuellement adaptés que pour des objets à très gros grains [BEVER 93] et ne prennent pas en charge la persistance. C'est pourquoi à la date de commencement de nos travaux, nous avons choisi de ne pas utiliser un tel système pour supporter notre plate-forme de coopération distribuée. Nous avons cependant prévu son intégration future sur une plate-forme à objets distribuée que nous avons décidé de concevoir dans le cadre du projet OSACA¹⁷ (Object System Architecture for Cooperative Applications).

¹⁰Open Software Foundation

¹¹Distributed Computing Environment

¹²Interoperable Systems Organisation

¹³Open Distributed Processing. Nous reviendrons sur le modèle O.D.P. dans la partie Analyse et Conception.

¹⁴Object Management Group

¹⁵Object Management Architecture

¹⁶COmmon Request Broker Architecture

¹⁷Le projet OSACA est un des projets du programme de recherche régional Ganymède portant sur les Activités Coopératives et la Communication Avancée (Cf. annexe).

4. RESUME

**LES BASES A OBJETS SONT UTILES POUR GERER LES
FEDERATIONS**

**LES BASES A OBJETS SONT UTILES POUR GERER LA
PERSISTANCE DES ESPACES DE COOPERATION**

**LE PRINCIPE D'ATOMICITE DOIT ETRE RELAXE POUR
SUPPORTER LES LONGUES SESSIONS CARACTERISTIQUES DU
TRAVAIL COOPERATIF**

**LES PRINCIPES D'ISOLATION ET D'ATOMICITE DOIVENT ETRE
RELAXES POUR PERMETTRE LA COLLABORATION**

**LA GESTION DE LA COLLABORATION DOIT ETRE SEPARÉE DE LA
GESTION DE LA PERSISTANCE QUAND LA GRANULARITE DU
PARTAGE EST INFERIEURE A LA GRANULARITE DE LA COHERENCE**

**UN OBJET EST UNE ENTITE PRIVILEGIEE DE GESTION DE LA
COLLABORATION**

**LES BASES A OBJETS CONSTITUENT LA MEILLEURE SOLUTION
ACTUELLEMENT**

**LES SYSTEMES DISTRIBUES A OBJETS CONSTITUENT LA
MEILLEURE SOLUTION POUR L'AVENIR**

CHAPITRE 6

DE LA COMMUNICATION ASSISTEE PAR ORDINATEUR A LA COLLABORATION INFORMATISEE

Nous développons ce chapitre sous la forme d'un historique. Notons que cet historique n'est pas forcément chronologiquement exact mais il permet d'introduire les différentes notions des systèmes de communication. Il part des systèmes les plus élémentaires en terme de support de la coopération pour arriver aux systèmes les plus évolués.

1. L'ECHANGE PRIVE D'INFORMATION

La première étape est la reproduction informatique du courrier postal. Le courrier électronique est une utilisation individualiste des moyens de communication rapide des réseaux informatiques, en particulier de l'Internet par la communauté scientifique.

A ce stade, on exploite simplement le système de transfert de messages, ce qui correspond au M.T.S. (Message Transfer System) de la norme X400.

C'est l'étape de **l'échange privé d'information**.

L'outil typique est sendmail sur UNIX™.

2. LA COMMUNICATION ET LA REPETITION INDIVIDUELLES DE L'INFORMATION.

La deuxième étape correspond à une adoption locale du courrier électronique : l'information que l'on reçoit ou que l'on envoie est communiquée au cercle de connaissances. On met une personne en copie d'un message envoyé ou on propage un message reçu vers une autre personne.

C'est ici essentiellement une amélioration de l'agent utilisateur, c'est à dire du composant qui assure l'interface entre l'utilisateur et le système de transfert. Ce composant est appelé User Agent (U.A.) dans la norme X400.

C'est l'étape de **la communication et la répétition individuelles de l'information**.

Les applications typiques sont les champs cc (copie), bcc (copie cachée) et forward (transmission) des en-têtes des messages électroniques.

3. LE TRI INDIVIDUEL DE L'INFORMATION

La troisième étape est une adoption étendue. Le courrier devient une source d'information, de connaissance. On utilise des répertoires pour classer les courriers lus. On a besoin d'outil pour rechercher des messages que l'on avait lus, en fonction de l'expéditeur, du sujet et du contenu.

Il s'agit encore d'une amélioration de l'agent utilisateur.

Cette étape introduit **le tri individuel de l'information**.

Les applications typiques sont les logiciels utilisateurs d'accès au courrier électronique, tels que mailtool sous OpenWindows™.

4. LE FILTRAGE INDIVIDUEL DE L'INFORMATION REÇUE.

La quatrième étape est une conséquence de la surcharge des utilisateurs. Pour répondre aux difficultés d'inondation, on crée des délégués informatiques, appelés agents. Le but est de filtrer la liste des messages avant consultation par l'utilisateur. Les études de MacKay montrent que le système est détourné pour filtrer après lecture car les utilisateurs craignent de perdre une information pertinente.

Ces agents interviennent au niveau de l'agent utilisateur.

Cette étape introduit **le filtrage individuel de l'information reçue**.

Les exemples d'application sont les agents d'Information Lens [MALON 87] et OVAL [MALON 92].

5. LA DIFFUSION DE L'INFORMATION VERS DES GROUPES.

La cinquième étape est due à l'identification de listes ou de groupes de destinataires. On crée des listes de distribution. Le courrier électronique peut même commencer à s'intégrer dans l'organisation en constituant la liste des membres d'un département, d'un laboratoire ou d'un projet.

Il peut s'agir simplement d'une amélioration de l'agent utilisateur qui remplace dans les champs destinataire et copie les noms des listes par leur contenu, avant de confier le message à l'agent de transfert.

Une exploitation plus sophistiquée des listes de distribution (par exemple, l'imbrication de listes) nécessite la prise en charge de l'expansion des listes par les agents de transfert.

Cette étape est marquée par **la diffusion de l'information vers des groupes**.

La technique des alias et des listes de distribution sont des applications de cette étape.

6. LE PARTAGE DE L'INFORMATION PAR UNE COMMUNAUTE.

La sixième étape introduit la notion de communauté. Au lieu d'envoyer des informations à une personne, on participe à un forum (ou une conférence). Les messages sont stockés pour la communauté, regroupés par thèmes et organisés en chaînes de discussion.

Cette étape introduit un composant de stockage entre certains agents utilisateurs et le système de transfert de messages. Ce composant est le *Messages Store* (MS) de la norme X400. Cette étape introduit également l'anonymat : avant cette étape, on envoyait des messages à des personnes que l'on connaissait mais, à partir de ce stade, l'utilisateur doit avoir un moyen de trouver l'adresse des forums qui pourraient l'intéresser. Dès lors, un autre service de communication devient très utile : le service de répertoire, tel que celui proposé par la norme X500.

Cette étape est caractérisée par **le partage de l'information par une communauté.**

Cette étape est supportée en télématique par les téléconférences et en informatique par les *News* et les *Bulletin Board Systems* de type *First Class*.

7. LE PARTAGE D'UNE BASE DE MESSAGES PAR UN GROUPE.

La septième étape consiste en l'exploitation de bases de messages. Plusieurs types d'agents sont connectés à une base de messages. Les agents déposants constituent une interface avec les différents systèmes de distribution du courrier : ils reçoivent les messages et les rangent dans la base. Les agents d'archivage ont pour rôle d'archiver les messages répondant à certains critères, précisés par l'administrateur de la base de messages. Des agents d'exploration offrent une interface interactive aux utilisateurs pour consulter directement la base de messages. Chaque agent filtrant est le représentant d'un utilisateur et sélectionne pour son compte dans la base les messages, selon les critères qu'il lui a fixés. Il y a donc filtrage des messages avant réception.

Il s'agit ici d'une amélioration du composant de stockage de messages, au niveau duquel les agents déportés interviennent mais aussi de l'agent utilisateur avec lequel certains agents déportés sur le composant de stockage interagissent.

Le partage d'une base de messages par un groupe caractérise cette étape.

Les filtres de Terry [TERRY 91] illustrent cette étape.

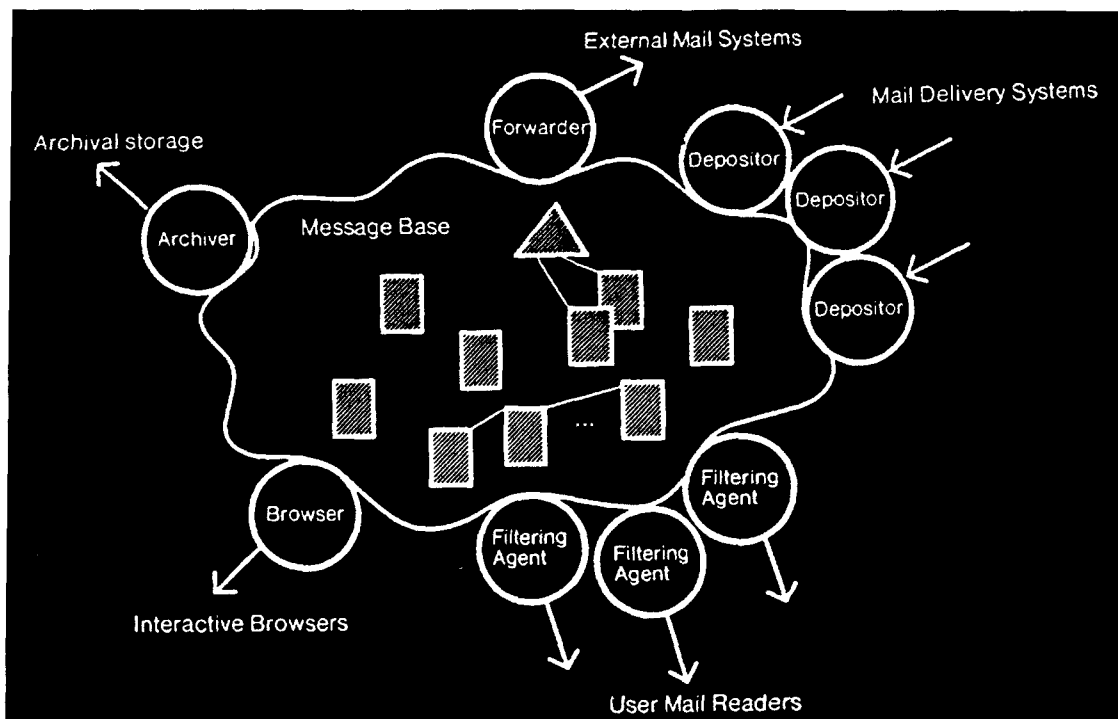


Figure 28. Les filtres de Terry

8. LA CIRCULATION DE L'INFORMATION DANS UNE ORGANISATION.

La huitième étape correspond à l'informatisation de la communication de groupe dans les bureaux. Le système d'aide à la communication est adapté à l'organisation. Le service de répertoire distribué est adapté à la structure de l'organisation.

Une première approche prend en compte uniquement les rôles structurels des membres de l'organisation : c'est l'approche orientée chemin. On analyse la hiérarchie de l'organisation pour édicter des règles d'adressage telles que "un rôle R1 ne peut adresser un message à un rôle R2 que s'il le met en copie à un rôle R3" ou "un rôle R1 ne peut jamais envoyer de message à un rôle R4". Cette première approche ne prend pas en compte les activités de l'organisation. On peut réaliser un tel système en incorporant ces règles aux agents utilisateurs. Le répertoire distribué X500 peut être utilisé pour stocker l'information sur les participants et leurs rôles.

Deux autres approches de l'aide à la communication de groupe prennent en compte les activités de l'organisation en terme de circulation de l'information.

L'approche orientée procédure analyse la circulation des documents entre les acteurs de l'organisation et en déduit des règles de routage des documents. Le système une fois installé est adapté au mode de travail de l'entreprise et doit pouvoir être modifié aisément si les procédures sont modifiées. LinkWorks™ [DIGIT 94] est un exemple de système récent orienté procédure. Ces règles de circulation peuvent être

centralisées au niveau d'un composant de stockage ou être distribuées sur les agents utilisateurs. X500 peut être utilisé pour stocker l'information concernant les participants et leurs rôles et les activités de l'organisation.

L'approche orientée formulaire analyse la structure des documents eux-mêmes et identifie des types de champs. En fonction des personnes qui sont habilitées à compléter les différents champs, on définit des rôles fonctionnels. On associe des règles à chaque champ : quel rôle fonctionnel peut le remplir, comment il peut être rempli, quel champ doit être rempli ensuite (éventuellement en fonction de la valeur donnée aux champs précédents). On associe des règles d'association entre les rôles structurels dans la hiérarchie de l'organisation et les rôles fonctionnels dans les procédures de traitement des formulaires. Un exemple connu dans la communauté du CSCW est Pages [HAMMA 87]. Dans les organisations, l'exemple le plus connu est Lotus Notes™. Cette approche nécessite une modification du composant de stockage qui doit incorporer des agents d'analyse des formulaires. X500 peut être utilisé pour stocker l'information concernant les participants et leurs rôles.

La circulation de l'information dans une organisation caractérise cette étape. Cette approche est connue sous le nom de *WorkFlow*.

9. LA CONVERSATION DE GROUPE : UNE ACTIVITE DU GROUPE.

La neuvième étape est le support de la conversation en tant qu'activité de groupe. Dans sa modélisation des organisations¹, ORDIT [DOBSO 91] identifie deux sortes d'actes : les actes instrumentaux et les actes de parole. On distingue l'espace de conversation organisant les éléments de la conversation (messages) et l'espace d'information qui contient les objets de la conversation. Les actes instrumentaux manipulent l'information. Un message typé matérialise un acte de parole. Un message ne contient plus qu'un acte de parole et des références à des objets de la conversation. Parallèlement à un système d'aide à la conversation de groupe, on peut utiliser un système hypermédia multi-utilisateurs. Il favorise l'organisation de l'espace d'information mais il faut prendre garde à ne pas utiliser les annotations publiques que l'on peut parfois lier aux documents comme des éléments de conversation. En effet, les systèmes hypermédia ne permettent pas de structurer l'espace de conversation qui se trouve éparpillé dans les annotations des noeuds. Les systèmes de support de la conversation proposent de structurer l'espace de conversation, orthogonalement à

¹L'analyse de organisations proposées par ORDIT est présentée plus loin dans cette thèse

l'espace d'information et d'établir des liens de référence de l'espace de conversation vers l'espace d'information.

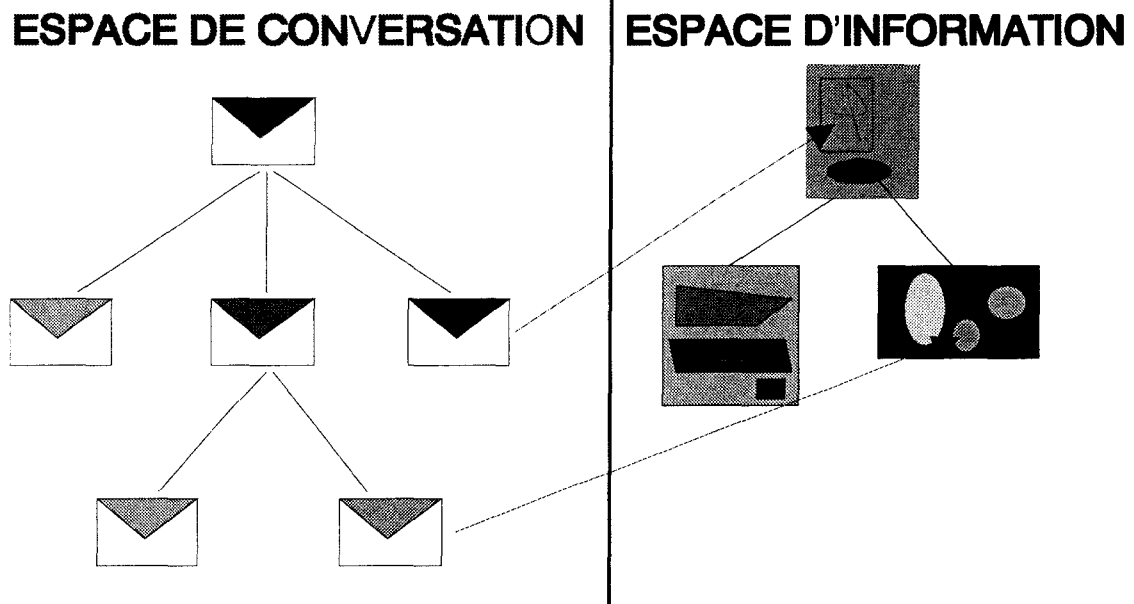


Figure 29. Séparer et Relier les Espaces de Conversation et d'Information

Le support des activités de conversation se fonde sur la théorie de la parole [AUSTI 62], et plus particulièrement du langage action [SEARL 82][WINOG 86] et la théorie de la conversation [LEPOU 93].

La théorie du langage action s'intéresse aux actes illocutoires (engagements et invitations à l'engagement). Winograd et Flores [FLORE 88], par exemple ont constitué une typologie des actes de langage action et modélisé des conversations par des réseaux de transitions entre les types d'actes locutoires. D'après Malone [MALON 87], grâce à la restriction des types de messages par un ensemble bien défini, on peut fournir un support de saisie d'information substantiel aux membres de l'équipe de collaboration.

La théorie de la conversation définit des types de conversation. La conversation pour l'action permet d'échanger des engagements. La conversation pour l'argumentation permet de soutenir ou de contredire des propositions. La conversation pour les possibilités permet au groupe d'explorer, de comparer des réponses à une question. La conversation pour l'éclaircissement offre de généraliser, spécialiser, illustrer une idée pour la clarifier. Chaque conversation définit en fait un type d'échange par un ensemble de types de messages et des règles de réponse pour chaque type de message. Ceci permet d'obtenir un graphe dirigé acyclique de la conversation qui donne aux membres du groupe une vision structurée de la conversation.

On peut réaliser de tels systèmes en répartissant les règles entre les composants de stockage et les agents utilisateurs. En effet, on peut distinguer deux types de règles. Les règles de conversation prennent en compte le rôle de l'utilisateur et le type du message auquel il veut répondre pour lui proposer des types de réponses possibles : elles peuvent donc être distribuées sur les agents utilisateurs. Les règles de médiation servent à déterminer le nouvel état global et l'évolution des rôles des participants suite à l'émission d'un message. Elles prennent en compte l'état global de la conversation, le type et la valeur les champs du nouveau message ainsi que le type et la valeur des champs du message auquel le nouveau message répond.

Par ailleurs, l'I.S.O. et le C.C.I.T.T. travaillent actuellement au développement d'un standard pour la communication asynchrone de groupe [SMITH 91] [BENFO 92]. La première version devrait contenir un modèle d'information de la communication et un standard pour les conférences, fondé sur ce modèle. Le modèle d'information est orienté objet. Les classes sont reliées par une hiérarchie d'héritage. Quatre classes génériques sont identifiées : Item, Entité, Domaine et Lien. Un item représente une unité d'information créée et échangée dans le cadre d'une application. Une entité représente une personne qui utilise une application. Un domaine regroupe des entités et des items. Les liens représentent des relations entre les objets, par exemple l'appartenance d'un item à un domaine comme l'appartenance d'un message à une conversation.

Cette dernière étape est l'étape de **la conversation de groupe**.

La Communication humaine est plus que la transmission d'information.

Il faut assister et mémoriser la structuration de la conversation.

Il faut à la fois séparer et relier les espaces de Conversation et d'Information.

Il faut relier les Activités de Conversation et les Activités Instrumentales.

CHAPITRE 7

LA COOPERATION: UNE QUESTION D'AGENTS

1. DES AGENTS INTELLIGENTS OUVERTS AU MONDE

L'évolution récente de l'intelligence artificielle montre l'importance qu'il y a à remplir les différents enjeux que nous avons identifiés pour supporter la collaboration, si l'on veut optimiser l'utilisation d'une intelligence partagée entre des **Agents**.

1.1. La résolution de problème distribuée

La **coopération** apparaît dans le champ de l'Intelligence Artificielle lorsque, à la fin des années soixante-dix, naît la recherche académique en **Intelligence Artificielle Distribuée**, sous la forme de son premier champ : la **Résolution de Problème Distribuée (R.P.D.)**.

L'idée est d'accélérer la résolution des problèmes en exploitant les capacités de parallélisme et la connexion grandissante des systèmes informatiques.

*Au lieu de se concentrer sur les techniques de résolution de problème, la R.P.D. repose sur des environnements où... des **modules coopèrent** en résolvant différentes parties du problème (des tâches), ou entrent en compétition pour appliquer différentes techniques de résolution de problèmes pour la même tâche. C'est de la responsabilité du contrôle de promouvoir la coopération et de gérer la compétition [CARDO 93].* La R.P.D. s'intéresse avant tout à la conception et la réalisation d'**architectures**. La coopération n'est pas un but à atteindre, pour lequel il faudrait optimiser les architectures, mais une contrainte générée par la distribution des modules. Les modules ne gèrent pas leur coopération avec les autres modules : c'est l'architecture qui organise les modules et qui définit le flux de contrôle et de communication entre les modules. La coopération se limite dans le cas de la R.P.D. à la coordination externe des modules. Les modules ne sont autonomes que dans la mesure où ils sont capables d'appliquer seuls la technique de résolution pour laquelle ils ont été conçus. Mais ils n'ont aucune autonomie vis-à-vis des buts qui leur sont fixés par le système de contrôle. La R.P.D. ne favorise pas la réutilisabilité, car les modules sont en fait très dépendants de l'architecture pour laquelle ils sont prévus et n'ont pas de capacité d'adaptation à d'autres architectures ou d'autres modèles.

1.2. Les systèmes multi-agent

Un nouveau courant de L'Intelligence Artificielle Distribuée est apparu au milieu des années quatre-vingts : les **Systèmes Multi-Agent**. Cette approche met l'accent sur l'**autonomie réelle** des modules, qui deviennent des **Agents**.

Il est alors extrêmement intéressant d'analyser les conséquences de cette autonomie sur les capacités à offrir aux agents pour leur permettre de coopérer. En effet nous retrouvons des caractéristiques que nous avons identifiées dans notre premier chapitre pour assister la coopération humaine, en particulier des caractéristiques liées aux enjeux de coordination et de communication.

D'abord, les agents doivent avoir les moyens de *percevoir les changements de l'environnement provoqués par des interventions humaines ou les actions d'autres agents* [Ibid.]. Cette capacité correspond aux caractéristiques 4 (partager des espaces d'actions), 11 (connaître le contexte des tâches) et 24 (partager une base d'information), que nous avons identifiées pour le support des enjeux de la collaboration. Dans le cas de la coopération humaine, en plus de participer à la satisfaction de l'enjeu de coordination, ces caractéristiques servent également à remplir l'enjeu de cohésion.

Ensuite, les agents nécessitent *des capacités de communication non seulement pour échanger des données mais aussi des connaissances et des plans* [Ibid.]. Cette idée est à rapprocher de la distinction entre l'information et la communication, que nous rappelons dans le chapitre 1 (page 8). Elle est également à rapprocher de l'évolution des systèmes de support de la communication, développée dans le paragraphe précédent intitulé "De la communication assistée par ordinateur à la coopération informatisée". Ces capacités correspondent aux caractéristiques que nous avons identifiées 6 (structurer la conversation), 13 (prendre des décisions en commun), 14 (prendre et obtenir des engagements) et 15 (connaître l'activité des autres).

De plus, les agents doivent avoir *des capacités de raisonnement social : un agent doit pouvoir raisonner à propos des activités des autres agents. Ceci est accompli par une représentation interne des attributs des autres agents (capacités, mécanismes de raisonnement, performance, etc.)* [Ibid.]. Il n'est qu'à comparer ces capacités aux caractéristiques 7 (Informations sociales et organisationnelles) et 10 (connaître les rôles) que nous avons identifiées pour le support de la coopération pour comprendre que les systèmes multi-agents rencontrent des préoccupations similaires à celles des systèmes de support de la coopération de groupe.

Enfin, les agents ont besoin de *capacités d'action* : un agent doit pouvoir agir sur son environnement en résultat de ses activités de résolution de problème. Ces capacités sont à rapprocher de la caractéristique 4 (partager des espaces d'action).

Les agents interagissent selon un protocole. Certains protocoles actuels utilisent des primitives de haut niveau, fondées sur la **Théorie du Langage Action**.

En conclusion, les capacités identifiées pour permettre à des agents autonomes de coopérer dans le cadre de l'intelligence artificielle distribuée, et en particulier des systèmes multi-agent, rejoignent les caractéristiques que nous avons retenues pour supporter la coopération de groupe. Le choix de la mise en oeuvre de la théorie du langage action est aussi commune au support de la coopération entre humains et entre agents artificiels.

2. LES AGENTS : DES FANTOMES D'UTILISATEUR DANS LA MACHINE

"Intelligent and cooperative systems" (ICIS) denote a new generation of information systems. ICISs are socio-technical systems which support cooperation in large distributed multi-agent settings. Agents can be humans cooperating with humans, humans cooperating with and through machines, and machines operating more or less autonomously.

[JARKE 93]

Tant que le support de la coopération par l'informatique s'est limité au développement de logiciels isolés conçus pour aider ponctuellement un petit groupe d'informaticiens à réaliser ensemble une tâche précise, la recherche en Systèmes d'Information n'était pas concernée.

Mais quand, comme aujourd'hui, et surtout en Europe, assister la coopération au moyen de l'informatique c'est concevoir des environnements de coopération à l'échelle d'une organisation, il est clair que l'on entre dans le domaine des Systèmes d'Information. En intervenant dans ce domaine, le CSCW modifie la conception même des Systèmes d'Information¹. Il s'agissait avant de concentrer l'analyse sur les données manipulées et leur traitement, désormais il faut s'intéresser en priorité aux humains et leurs activités. Or ces activités sont essentiellement des activités de communication de groupe. L'entité centrale du schéma du Système d'Information devient alors l'**agent humain**, en tant que membre d'une organisation et participant à de nombreuses activités. Il ne s'agit plus de stocker les caractéristiques (coordonnées, situation de famille, poste, salaire) d'un membre de l'organisation dans la base de

¹Cet aspect est développé dans une autre partie de cette thèse. Nous développons aussi cette idée dans [DERYC 95].

données de gestion du personnel. Il s'agit bel et bien de considérer les humains comme des agents actifs, avec lesquels il faut coopérer et qui utilisent le système pour coopérer entre eux. Le système d'information doit jouer le rôle d'assistant dans les relations de coopération d'un humain avec ses collègues : il mémorise pour eux leurs rôles dans les différentes activités, les actions qu'ils ont accomplies, les engagements qu'ils ont pris et ceux qu'ils ont reçus, l'état des activités en cours, etc. Le système d'information doit en définitive gérer une organisation virtuelle complète. L'humain n'est plus pris en compte à la périphérie du système, juste au niveau de l'interface Homme-Machine mais il a ses **agents représentants**, ses fantômes pourrait-on dire, au coeur du système d'information. Cette idée essentielle est développée plus loin dans ce document, dans la partie IV concernant l'analyse et la conception de Co-Learn.

3. L'AGENT : UN ELEMENT CLE POUR L'ADAPTATION DES INTERFACES HOMME-MACHINE A LA COOPERATION

La coopération à distance médiatisée par ordinateur fait l'objet de recherches dans différents domaines de l'Interface Homme Machine. Elle définit de nouveaux défis pour l'ergonomie, la conception des architectures et l'évaluation des interfaces. Elle tisse des liens avec les interfaces multimodales et les interfaces nouvelles. L'importance que prend le travail coopératif dans le domaine des interfaces Homme-Machine se reflète dans la naissance et la croissance de l'atelier collectif aux journées IHM [TELEC 92] [ECL 93, pp. 65-122] [TRIGO 94, pp. 133-210].

La **métaphore** du bureau, actuellement un standard des Interfaces, est une métaphore individualiste du travail, qui représente l'espace de travail dans un espace à deux dimensions. Elle nécessite d'être modifiée pour prendre en compte les deux dimensions manquantes pour la prise en compte du travail de groupe : la troisième dimension spatiale est nécessaire pour représenter la localisation de plusieurs participants dans un espace de travail commun (Cf. [DERYC 93]), la dimension temporelle doit trouver une représentation pour intégrer la téléprésence. De nombreux travaux préconisent l'utilisation de la métaphore de **Room** (Pièce Virtuelle) pour prendre en compte ces dimensions. Nous développons une analyse du développement de la métaphore de *Room* dans la communauté des IHM, puis du CSCW dans le chapitre seize de cette thèse.

La conception d'interfaces multi-utilisateurs a aussi des conséquences sur l'**architecture** des interfaces et réclame des mécanismes de communication par événements (ou envois de message). Les applications coopératives bien structurées doivent séparer clairement noyau fonctionnel de la présentation de sorte que plusieurs

présentations puissent être connectées au même noyau fonctionnel, éventuellement par l'intermédiaire de contrôleurs de coopération qui filtrent et distribuent les événements. Certaines interfaces multi-utilisateur permettent ainsi de partager des applications qui n'ont pas été conçues pour être partagées, mais ont été structurées en séparant la partie présentation de la partie abstraction. Les **modèles d'interface multi-agent** tels que PAC [COUTA 90], MVC et ALV, proposent une structuration des objets interactifs qui peut être adaptée pour prendre en compte la dimension coopérative [CROIS 95]. Dans le chapitre 9 de cette thèse, nous décrivons l'intégration de l'architecture des interfaces dans un environnement distribué de support de la collaboration. Le lecteur intéressé par une analyse détaillée de l'impact de la coopération sur les interfaces hommes-machines pourra aussi consulter la thèse de Pascal Croisy [CROIS 95]. D'autre part, [BENTL 94] présente une synthèse très intéressante des enjeux des interfaces multi-utilisateurs.

4. LES AGENTS COMMUNIQUENT EN GROUPE

Les mécanismes de communication entre processus en mode point à point se révèlent inadaptes au support de la coopération, qui nécessite des mécanismes de communication de groupe. Les premiers travaux dans le domaine de la communication de groupe n'avaient pas pour objet de supporter la coopération humaine. L'objectif de certains travaux était d'augmenter la tolérance aux pannes, en demandant à plusieurs unités de traitement indépendantes de réaliser la même tâche. D'autres travaux proposent d'augmenter la vitesse de traitement en plaçant plusieurs unités de traitement en compétition et en utilisant la première réponse qui arrive du groupe. On peut également vouloir utiliser la communication de groupe pour augmenter la fiabilité des systèmes de prise de décision : en posant la même question à plusieurs agents intelligents raisonnant de façon différente et en exploitant la liste des réponses.

Les besoins spécifiques de la communication de groupe se regroupent en plusieurs catégories :

- Gestion et exploitation du répertoire dynamique des participants, prise en charge de la diffusion,
- Multimédia : gestion de flux parallèles ou temps réel, flux synchrones, flux isochrones,
- Facteur d'échelle : utilisation de relais pour la minimisation de la bande passante et la minimisation des coûts,

- Qualité de la diffusion : l'atomicité de la diffusion garantit que tous les participants recevront les messages diffusés dans le groupe dans le même ordre.

Un système de gestion de communication de groupe doit :

- permettre la création dynamique de groupes,
- gérer la diffusion atomique de messages à l'ensemble des membres d'un groupe (sérialisation, synchronisation de l'envoi et de la réception des messages entre tous les membres du groupe),
- accompagner les arrivées et départs des membres du groupe.

La mise en oeuvre des systèmes de communication de groupe font actuellement l'objet de recherches spécialisées. ISIS [BIRMA 90] est un exemple de système de communication de groupe. De récents travaux [BEAUD 92] tendent à montrer que la mise en oeuvre de ISIS est encore trop lourde pour l'intégrer dans les collecticiels actuels.

5. LA METAPHORE D'AGENT DANS CO-LEARN

Le paradigme d'Agent que nous employons [BARME 95] [CROIS 95a] [CROIS 95b] [DERYC 95] est proche du paradigme d'Objet. Il est tout à fait semblable au paradigme d'Agent des systèmes multiagent, au sens des interfaces Homme-Machine, c'est à dire tel qu'il est décrit dans [COUTA 91] :

Un agent est un système de traitement de l'information : il possède des récepteurs et des émetteurs pour acquérir et produire des événements ; il dispose d'une mémoire à deux niveaux : l'une pour enregistrer les événements détectés par les récepteurs, l'autre pour mémoriser un état ; il comprend un processeur cyclique spécialisé dans le traitement d'une ou plusieurs classes d'événements. Le résultat d'un traitement se traduit généralement par un changement d'état de l'agent et par l'émission de nouveaux événements. L'état fournit à tout instant l'image de l'évolution d'un agent.

Le modèle multiagent se caractérise par une organisation fortement modulaire, des traitements exécutés en parallèle et une communication par événements.

Cependant, nous abusons parfois du terme Agent, en l'employant pour des composants qui n'ont pas toutes ces caractéristiques. Nous nous permettons cet abus quand nous voulons insister sur la notion de représentation : l'Agent est alors synonyme de Délégué, de Représentant, de "Proxy"² intelligent. Cette représentation est assortie d'une propriété d'autonomie : contrairement à un Proxy, un Agent est

²La notion de *proxy* est souvent utilisée dans les systèmes réalisant la distribution de Smalltalk. Cf. [DECOU 86], [VIEVI 90b], [DOLLI 91] et [VILER 92]. C'est Shapiro [SHAPI 86] qui introduisit les *proxies* comme représentants locaux de services distants.

capable d'interpréter les événements que lui transmet celui qu'il représente. Cet abus de la notion d'Agent nous tente également beaucoup lorsque nous parlons d'Objet pour signifier qu'il ne s'agit pas d'un objet passif, uniquement dans son sens structurel, tel qu'on en trouve dans certains systèmes de persistance, mais bien d'un objet "à la Smalltalk" [GOLDB 84], responsable de son comportement. Nos faux Agents sont souvent en définitive de vrais Objets.

**PARTIE IV
ANALYSE, CONCEPTION ET DEVELOPPEMENT DE CO-LEARN**

TABLE DES MATIERES DE LA QUATRIEME PARTIE

CHAPITRE 8 SPECIFICATIONS : DEVELOPPER UN RESEAU EXPERIMENTAL D'APPRENTISSAGE COOPERATIF A DISTANCE	100
1. Les spécifications informelles : un environnement pour l'enseignement ouvert.....	100
2. Prise en compte des contraintes	104
3. Présentation générale du système proposé : Co-Learn.....	105
4. La métaphore de la Salle Virtuelle ou Room	107
5. La métaphore d'Outil et le paradigme d'Activité.....	109
 CHAPITRE 9 LE CADRE DE L'ANALYSE	 110
1. Les méthodes d'analyse considérées dans notre étude	110
1.1. ODP : Analyser les systèmes distribués ouverts.....	110
1.2. MOCCA : Analyser les environnements distribués ouverts de CSCW	110
1.3. ORDIT : Déterminer les besoins de l'organisation en matière de Système d'Information pour le CSCW.....	112
1.4. AMIGO : Concevoir les systèmes de coordination des activités de communication de groupe.....	115
1.5. [ELLIS 94] : Permettre aux utilisateurs d'analyser les collecticiels	117
1.6. [HENNE 94] : Intégrer les collecticiels par le partage d'un modèle commun de l'entreprise.....	118
2. Articulation des approches.....	119
2.1. Contributions au point de vue de l'ORGANISATION.....	119
2.2. Contribution aux points de vue de l'INFORMATION.....	123
2.3. Contribution aux points de vue DYNAMIQUES	126
2.4. Contribution aux points de vue de l'INGENIERIE.....	128
2.5. Contribution aux points de vue de la TECHNOLOGIE.....	129
2.6. Influence de MOCCA sur ODP.....	130
2.7. Articulation de MOCCA, ORDIT et AMIGO.....	131
3. Notre proposition : un environnement de conception, de développement et d'intégration d'activités de coopération.....	132

CHAPITRE 10 LES POINTS DE VUE DE L'ORGANISATION 133

1. La base de la pyramide : les situations d'apprentissage coopératif	134
1.1. Composer des environnements de coopération	134
1.2. Concevoir les activités de coopération	141
2. Le deuxième niveau de la pyramide : la gestion des unités organisationnelles.....	179
3. Le sommet de la pyramide : la gestion générale de l'organisation.....	181
4. Point de vue de l'organisation de Ellis : les vues offertes a l'utilisateur	183
4.1. Vues sur les objets	183
4.2. Vues sur les participants	183
4.3. Vues sur le contexte	185

CHAPITRE 11 LES POINTS DE VUE DE L'INFORMATION 188

1. Le modèle ontologique.....	188
1.1. le domaine organisationnel	189
1.2. Le domaine de la coopération en temps différé	190
1.3. Le domaine de la documentation	192
1.4. Le domaine de la vue partagée.....	193
2. Le modèle conceptuel des données	193
2.1. Le domaine des Salles Virtuelles.....	195
2.2. Le domaine de la Communication	197
2.3. Le domaine de l'Organisation	201
2.4. Le domaine de l'Expertise Utilisateur	204
2.5. Le domaine de l'Information.....	205

CHAPITRE 12 LES POINTS DE VUE DYNAMIQUES 207

1. De l'application interactive personnelle au système d'information coopératif.....	207
1.1. Les applications interactives mono-utilisateur	207
1.2. Les systèmes de partage de vue	210
1.3. Les applications interactives de groupe	211
1.4. Les espaces de coopération.....	213
2. Architecture logicielle distribuée de l'environnement de coopération Co-Learn.....	217

CHAPITRE 13 LES POINTS DE VUE DE L'INGENIERIE	226
1. Programmation des composants du système distribué Co-Learn	226
1.1. Programmation des composants logiciels des clients	226
1.2. Programmation des composants logiciels des serveurs	226
2. Techniques de contrôle logiciel.....	227
2.1. Contrôle logiciel sur les clients.....	227
2.2. Contrôle logiciel sur les serveurs.....	228
3. Techniques de Communication.....	228
3.1. Techniques de Communication inter-applications sur les clients	228
3.2. Techniques de communication sur les serveurs.....	230
3.3. Techniques de communication entre les clients et les serveurs	231
4. Techniques de gestion de la persistance.....	232
CHAPITRE 14 LES POINTS DE VUE TECHNOLOGIQUES	235
1. Les serveurs Co-Learn	235
2. Les clients Co-Learn	236
3. Le réseau Co-Learn	237



CHAPITRE 8

SPECIFICATIONS : DEVELOPPER UN RESEAU EXPERIMENTAL D'APPRENTISSAGE COOPERATIF A DISTANCE

1. LES SPECIFICATIONS INFORMELLES : UN ENVIRONNEMENT POUR L'ENSEIGNEMENT OUVERT

L'autonomie des apprenants est une des raisons d'être de l'enseignement à distance. En général, les personnes qui suivent un enseignement à distance reçoivent une mallette pédagogique contenant des documents de travail (éventuellement multimédia) et une description de la méthode de travail. Elles font des exercices qu'elles envoient à un correcteur.

L'intérêt de cette méthode est de permettre de travailler où l'on veut, quand on le veut. Mais son principal inconvénient est l'isolement des apprenants. Or il est certain que l'on n'apprend pas seul : on ne peut comprendre et assimiler des connaissances objectives qu'en confrontant ses propres idées et ses interprétations à celles des autres. Dans le cas de la formation continue, l'échange de points de vue est à la fois plus enrichissant et plus nécessaire, à cause de l'expérience et des schémas de pensée que les apprenants ont acquis. Il est primordial de distinguer autonomie et isolement.

La plupart des apprenants qui utilisent l'enseignement à distance le font soit parce qu'ils y sont forcés par leur manque de disponibilité ou leur éloignement des centres de formation qui les intéressent, soit pour conserver une autonomie qu'ils n'ont pas dans l'enseignement traditionnel. Certains apprenants optent pour l'enseignement à distance délibérément dans le but d'être isolé des autres pour apprendre mais ils sont minoritaires et les résistances psychosociologiques qui les amènent à ce comportement pourraient disparaître dans le cadre d'une coopération à distance. Mieux encore : une expérience de l'apprentissage coopératif à distance peut leur permettre non seulement de coopérer pour apprendre mais également d'apprendre à coopérer.

Le C.U.E.E.P.¹ pratique depuis déjà longtemps l'enseignement ouvert. Les apprenants inscrits en formation à distance sont réunis dans des groupes. La formation comprend de longues périodes d'enseignement à distance traditionnel pendant lesquels les membres du groupe peuvent utiliser le fax et le téléphone pour communiquer avec les animateurs de formation. Durant ces périodes, les apprenants peuvent aussi se rendre dans le centre de ressources le plus proche pour utiliser les ordinateurs personnels et les didacticiels laissés à leur disposition. Ces périodes d'isolement sont ponctuées par des séances de regroupement au cours desquelles l'animateur peut vouloir donner un cours magistral ou organiser un atelier de travail afin que le groupe se livre à une étude de cas ou travaille à un projet commun. Les chercheurs de l'équipe OPEN de TRIGONE, laboratoire de recherche pluridisciplinaire du C.U.E.E.P., qui sont aussi les animateurs de ces formations à distance, ont identifié le besoin de faciliter et d'améliorer la coopération des apprenants d'un groupe de formation.

Quand on étudie le cycle d'apprentissage des apprenants dans le cadre d'un enseignement ouvert tel que décrit ci-dessus, on distingue quatre situations :

- Le cours magistral,
- L'atelier de travail en commun,
- Le travail isolé,
- L'assistance à distance.

¹ Institut de l'Université des Sciences et Technologies de Lille, spécialisé dans l'ingénierie de formation et l'organisation de dispositifs de formation innovants.

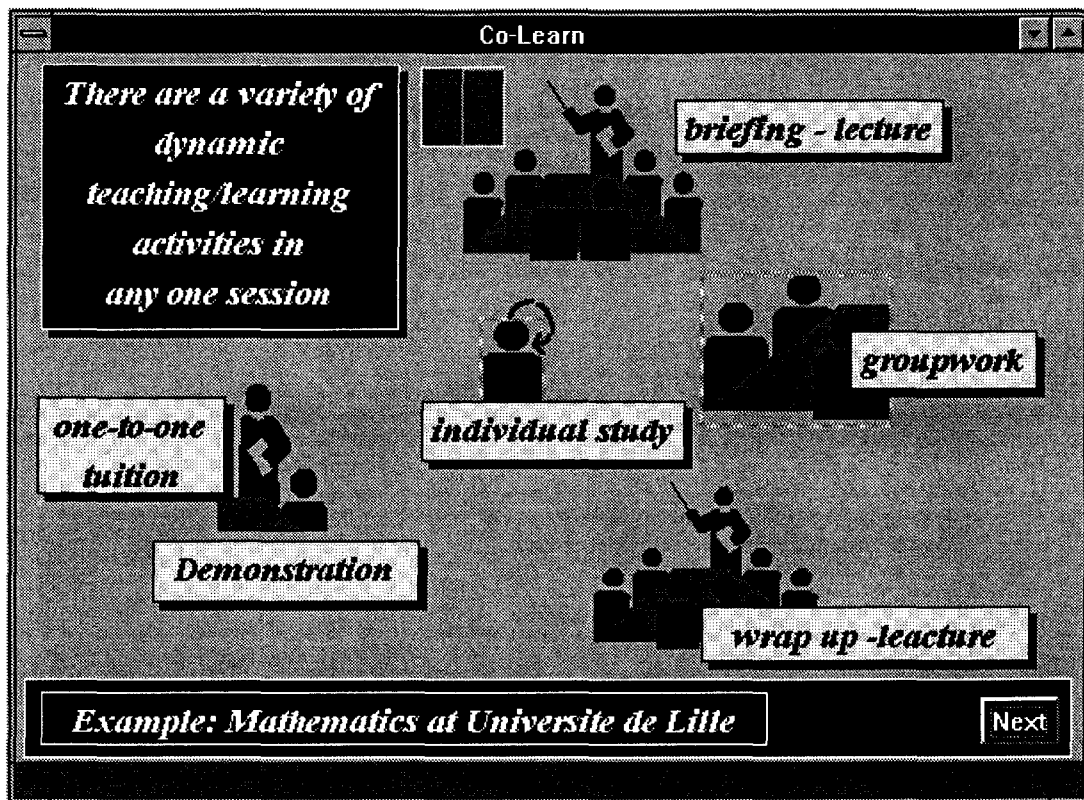


Figure 30. La variété des situations d'apprentissage [SMITH 94]

Nous identifions des besoins pour chacune de ces situations. D'abord, le cours magistral et l'atelier de travail en commun ne peuvent être mis en oeuvre que lors des séances de regroupement, qui nécessitent la réunion des participants et posent des problèmes d'organisation. Il serait intéressant de pouvoir les mettre en oeuvre à distance pour que les apprenants puissent y participer à partir de chez eux ou d'un lieu proche de chez eux, par exemple en se rendant au centre de ressources le plus proche.

Pour le cours magistral, un système tel que Télé-Amphi [SEGUI 92] permettant à plusieurs groupes distants de suivre l'exposé d'un enseignant semble bien adapté. Par contre, un système permettant beaucoup plus d'interaction est nécessaire pour les ateliers de travail.

Dans les situations d'assistance à distance, l'animateur et l'apprenant se sentent bien dépourvus avec pour seul moyen de communication le téléphone et le fax, en particulier quand l'apprenant rencontre une difficulté pendant l'utilisation d'un didacticiel. Le formateur aurait bien besoin de voir, désigner ou contrôler l'ordinateur de l'apprenant. Même dans une situation générale sans ordinateur, les interlocuteurs sont frustrés de ne pouvoir joindre le geste à la parole pour montrer et commenter en temps réel les objets de leur discours.

Les périodes de travail en mode isolé sont trop longues, parce que les séances de regroupement sont difficiles à organiser : un système permettant le regroupement à distance augmenterait la fréquence des réunions et diminuerait ainsi la durée des périodes d'isolement.

Par ailleurs, il serait intéressant de permettre aux participants de continuer à coopérer quand ils ne peuvent pas travailler ensemble au même instant. Un système de coopération de groupe asynchrone permettrait de poursuivre la coopération entre les séances de regroupement. On voit ici tout l'intérêt de l'intégration dans un même environnement des situations de regroupement et des situations de coopération asynchrone : on doit pouvoir accéder aux mêmes ressources dans les deux situations. Le travail effectué dans l'une des situations doit pouvoir être utilisé dans l'autre. Notons que le C.U.E.E.P. a déjà expérimenté des systèmes de coopération asynchrone [VIEVI 90a] [VIEVI 91] : il s'agissait de systèmes de messageries sur Minitel. Cette expérience a permis aux membres des équipes OPEN et NOCE du laboratoire TRIGONE d'identifier les inconvénients de la première génération de systèmes de coopération asynchrone et a été l'occasion d'une réflexion sur ce que devraient être les systèmes de nouvelle génération [DERYC 92]. Les principaux défauts des messageries télématiques étaient la pauvreté de l'interface (Minitel) et le manque de structuration des conversations et d'outils de navigation dans les conversations.

Une analyse des obstacles de l'enseignement ouvert montre également la difficulté des apprenants à appréhender leur situation par rapport à l'organisation de formation. Il serait donc utile que dans les différentes situations d'apprentissage auxquelles ils sont confrontés, les participants aient une notion de son contexte organisationnel. Cette intégration des informations organisationnelles est peut-être plus importante encore pour les formateurs, qui non seulement sont membres permanents de cette organisation mais en plus ont en charge la gestion de la partie de cette organisation que forment les groupes d'apprenants qu'ils animent. Les formateurs ont besoin de moyens d'administration et de supervision des groupes, de leurs ressources (documents et outils) et de leurs activités.

Voici donc tel qu'il était posé le cadre de notre travail. Il est à noter que beaucoup de ce qui a été dit à propos de l'apprentissage est également vrai pour le travail en général : le travail à distance doit offrir aux télé-travailleurs l'autonomie, sans provoquer leur isolement. De sorte qu'un environnement permettant l'apprentissage coopératif à distance devrait pouvoir être utilisé ou adapté pour supporter le travail coopératif à distance. Ceci est d'autant plus vrai qu'un

environnement devant supporter la coopération des groupes devra pouvoir être facilement ajusté, à cause de la variété des pratiques des groupes et en particulier des modes de conduite de groupe; il devra être également ouvert à la diversité des outils traditionnels des groupes et des individus.

C'est ainsi que le laboratoire TRIGONE a soumis une proposition de projet appelé Co-Learn² (pour Apprentissage Coopératif) à la Commission de la Communauté Européenne, dans le cadre du programme DELTA [HUYTO 92].

2. PRISE EN COMPTE DES CONTRAINTES

Suite à l'étude des spécifications informelles, de l'état de la technologie des réseaux de communication et des ressources du C.U.E.E.P. et des autres organismes susceptibles de mettre en oeuvre un système d'apprentissage coopératif à distance tel que Co-Learn, nous avons déterminé plusieurs contraintes.

- pour supporter le niveau d'interaction désiré, il était impossible d'envisager l'utilisation d'un réseau de communication asynchrone, tel que le réseau téléphonique commuté. Comme il apparaissait intéressant de pouvoir intégrer l'audio- ou la vidéoconférence aux postes de travail Co-Learn, il a été décidé d'utiliser le Réseau Numérique à Intégration de Services [DICEN 90]. Le choix du R.N.I.S. fut également influencé par l'intérêt que lui portait la Commission Européenne. Le coût des connexions **R.N.I.S.** et des installations de ligne interdisait l'installation des postes au domicile des apprenants. Il fut donc décidé qu'ils seraient mis à disposition des apprenants dans les centres de ressources qui seraient équipés de connexions R.N.I.S..

- parce que les apprenants sont en général novices en informatique et parce que les formateurs ont l'habitude d'utiliser ce type d'ordinateur, le PC fut choisi comme poste de travail. L'équipe NOCE décida de développer le logiciel des postes de travail sous **Windows 3.1**, bien que les formateurs fussent habitués à l'utilisation de logiciel d'Enseignement Assisté par Ordinateur sous le système d'exploitation MS-DOS et malgré la nécessité d'un apprentissage pour les formateurs et les apprenants aux systèmes de multi-fenêtrage. Cette décision se justifiait par la volonté d'exploiter les avantages de Windows en ce qui concerne ses possibilités de gestion de la mémoire, ses fonctionnalités multimédia, ses possibilités multitâches et ses nouveaux moyens de communication entre applications (DDE et OLE [HELLER 92]). Ce choix nous permettait également d'envisager l'évolution de Co-Learn vers les systèmes d'exploitation Windows for Workgroup et Windows NT.

² Le projet Co-Learn est présenté dans l'annexe I

- Les deux décisions précédentes en impliquent une troisième : on ne pouvait envisager de distribuer les objets persistants sur les postes de travail car on ne pouvait maintenir en permanence ni le fonctionnement de ces postes, ni les liens de communication avec eux. De plus, les postes de travail devaient être **banalisés**, puisque mis à la disposition de tous les apprenants mais **personnalisables** afin de pouvoir s'adapter à la fois à la situation d'apprentissage de son utilisateur, aux caractéristiques du groupe et de l'individu. Il fallait donc envisager une centralisation au moins partielle du système de persistance. La partie centrale du système pourrait se composer d'un serveur ou d'un ensemble distribué de serveurs interconnectés (semblable au réseau de serveurs télématiques) auxquels les postes de travail se connecteraient pour obtenir la configuration adaptée à l'apprenant qui les utilisent et lui proposer les différentes situations de travail actuelles ou planifiées de son groupe. Cette configuration transforme ainsi la station en une entité du système distribué.

3. PRESENTATION GENERALE DU SYSTEME PROPOSE : CO-LEARN

Le choix primordial à la conception du système Co-Learn a été celui de la métaphore que l'on proposerait à l'utilisateur pour représenter les différentes situations d'apprentissage : la **Salle Virtuelle** [DERYC 93] [HOOGS 93] [HOOGS 94]. Notre première approche a consisté à définir un type de salle pour chaque situation d'apprentissage. Pour chaque type de salle, nous avons déterminé un ensemble d'**Outils** [COLEARN 92b] [HOOGS 93] adaptés essentiels ou optionnels. Pour assurer la coordination des actions du groupe, nous avons conçu un système de gestion des **Rôles** et de **Médiation** des **Activités** [HOOGS 92] [VIEVI 94a] [VIEVI 94b]. Le principe de coopération du groupe était de participer à des **Activités instrumentales** ou des **Activités de Conversation** [VIEVI 93] par l'intermédiaire d'Outils. Chaque Outil devait pouvoir adapter son interface et son comportement selon l'évolution de l'Activité pour laquelle il serait utilisé et selon l'évolution du rôle que le participant jouerait dans cette Activité.

Ces métaphores d'interface ont pu être renforcées comme métaphores de conception et d'implémentation, grâce à l'utilisation de l'approche orientée objet à tous les niveaux de développement du système. Ainsi retrouve-t-on dans le système d'information du serveur les classes Room, Tool, Activity, Mediator et Role. La métaphore de Salle Virtuelle est également utilisée comme unité structurante du système de communication de groupe. Tous les postes de travail dont les utilisateurs sont dans la même Salle Virtuelle communiquent entre eux et avec le serveur par l'intermédiaire de la même Salle Virtuelle de communication. C'est la Salle Virtuelle

qui détecte les entrées et sorties des participants et en avertit les Activités et Outils concernés. C'est encore la Salle Virtuelle qui, à la demande des Activités, charge et adapte les Outils nécessaires sur les postes de travail. La Salle Virtuelle diffuse les actions des participants vers les Activités concernées et répercutent les évolutions des Activités vers les participants impliqués. La Salle Virtuelle sait résoudre les identités des Outils, des Activités et des Participants. La Salle Virtuelle a aussi pour rôle de réaliser la diffusion atomique des messages entre les participants.

Ainsi, nous verrons que notre système est fidèle, depuis la conception jusqu'à l'implémentation, à la métaphore de la Salle Virtuelle telle qu'elle est analysée dans les travaux les plus récents du domaine du C.S.C.W., comme en témoignent ces quelques extraits d'un article de Steve Benford [BENFO 93] :

We define a room as a virtual computer space within which people interact, either with each other or with the various tools which they find there. Put in a more abstract way, each room defines a particular domain of interaction. [...] Rooms also provide awareness of the presence and comings and goings of all occupants and tools. [...] The advantage of locating tools in rooms is that all tools become collaborative to some degree through the ether that surrounds them.

Pour répondre aux besoins d'**administration de l'organisation** et de **contextualisation des situations d'apprentissages** et des participants dans l'organisation, c'est encore la Salle Virtuelle qui a été utilisée comme unité élémentaire d'organisation. L'Organisation est structurée en **Unités d'Organisation** : il peut s'agir par exemple de départements, de services ou/et de projets dans une entreprise, de matières, d'unités de valeur, d'unité de formation ou/et de départements dans un institut de formation, une école ou une université. Les administrateurs de ces unités d'organisation peuvent créer dynamiquement des Salles Virtuelles. Lorsqu'un participant se connecte au système, on lui propose les différentes unités d'organisations (il peut s'agir par exemple de modules de formation) puis dans chaque unité les différentes Salles Virtuelles, auxquelles il a accès.

Les formateurs et les apprenants se voient affecter des **Rôles Structuraux** qui définissent leurs droits pour créer des unités d'organisation, y inscrire des membres et définir leurs Rôles Structuraux, ajouter des Salles Virtuelles dans une unité, y ajouter des Outils et des Participants dont ils peuvent définir les Rôles Structuraux et Fonctionnels.

Pour compléter cette présentation générale du système Co-Learn, il nous faut finalement présenter la notion d'**Outil Privé**. Les outils privés diffèrent des autres Outils de Co-Learn (que nous appelons Outils Partagés) par leur degré d'intégration

dans l'environnement. Ces outils privés sont constitués de toutes les applications disponibles sous Windows. Leur degré d'intégration devrait augmenter avec le développement de l'O.L.E., qui assurera l'interopérabilité de toutes les applications Windows et avec le développement de la notion d'interface de coopération (telle que celle de serveur D.D.E.). Le principe d'utilisation d'un Outil Privé est la réalisation en mode isolé d'un travail, que l'on présente au groupe ou que l'on intègre au travail du groupe par l'utilisation d'un **Outil Partagé** qui communique avec l'outil privé par un mécanisme d'échange de données (par exemple le presse-papiers ou l'O.L.E.). Les Outils Partagés sont des outils développés spécifiquement dans l'environnement Co-Learn pour s'intercaler entre le participant et une Activité du système via leur interface de coopération (réalisée par exemple grâce au mécanisme D.D.E.). Un Outil Partagé a le devoir de communiquer à l'Activité (plus précisément à son **Médiateur**, qui exécute les Règles de Médiation³) certaines actions demandées par le participant et il doit obéir à son **Agent** qui, respectant les Règles de Conversation associées au Rôle du Participant dans l'Activité, répercute sur l'Outil les modifications signalées par l'Activité.

4. LA METAPHORE DE LA SALLE VIRTUELLE OU ROOM

La première métaphore d'interface est la Salle Virtuelle, dérivée de la métaphore de "Room" de Stuart K. Card. D'abord inspirée de la métaphore de projet de Smalltalk [GOLDB 84], elle est conçue pour diminuer la surcharge spatiale dans les systèmes de fenêtrage [HENDE 86]. Mais plus encore, son intérêt est de diminuer la surcharge cognitive de l'utilisateur, en lui permettant de distribuer l'espace de ses tâches individuelles dans des espaces de travail virtuels (mais concrétisés). La Room a aussi l'avantage de faciliter la personnalisation ("tailorability") de l'environnement de travail de l'utilisateur. Cependant, cette métaphore dépasse rapidement son cadre initial. Dès 1987, Stuart K. Card et D Austin Henderson [CARD 87] y adjoignent la métaphore de catalogue : ils prennent ainsi en compte l'intégration des ordinateurs personnels sur les réseaux locaux. Un ordinateur personnel placé sur un réseau local peut accéder à beaucoup d'applications mais l'installation de ces applications est généralement une tâche fastidieuse. Le rôle du catalogue est justement de simplifier cette procédure d'installation en proposant à l'utilisateur d'ajouter les applications du catalogue dans les Salles Virtuelles de son choix : l'utilisateur n'a qu'à les désigner. En 1989, Card,

³Les règles de Médiation définissent les conséquences des actions des Participants sur l'Activité, en fonction de la phase de l'Activité. Les règles de Conversation définissent les droits du Participant en fonction de l'état de son Rôle et de la phase de l'Activité.

Robertson et Mackinlay proposent d'appliquer la troisième dimension et l'artefact du coprocesseur cognitif à la métaphore de Room.

En 1991, Card, Robertson et McKinley [CARD 91] présentent un nouveau concept pour l'interface utilisateur des systèmes de recherche d'information : le "Information Visualizer". Le but est toujours d'aider l'utilisateur à organiser son espace de travail personnel. Le dessin qui se trouve dans l'article montre d'ailleurs un utilisateur seul dans son bureau. Certaines photos d'écran dévoilent par contre des détails très intéressants: l'un des systèmes de visualisation permet d'explorer un bâtiment et de repérer les bureaux et leurs occupants. Pourtant l'article ne fait aucune allusion à l'application possible du système ou de la métaphore pour faciliter le travail coopératif.

Parallèlement, la métaphore de Room se développe dans le champ de recherche du travail coopératif : en 1989, Christian M. Madsen [MADSE 89] explique que la métaphore populaire du bureau n'est plus suffisante pour concevoir des systèmes supportant le travail coopératif. Il propose la métaphore de l'immeuble de bureaux. Comme il le dit, dès lors le concept de Salle Virtuelle devient central, et particulièrement la salle de travail de groupe. C. M. Madsen explique que la métaphore de Room de Xerox diffère de celle qu'il présente, malgré des similitudes. Effectivement, la Room telle qu'elle est utilisée dans le CSCW ne se contente pas de réunir des applications mais réunit avant tout des personnes et des informations. Elle est un espace partagé de communication et d'action. Cette métaphore de Room se propage rapidement dans la communauté CSCW. En fait, la notion de salle de réunion virtuelle apparaissait déjà en 1988 dans le système de conférence multimédia Rapport [RODDE 91], développé par les laboratoires AT&T Bell au New Jersey, ainsi que dans Cruiser [Ibid.] à Bellcore. Par la suite la Room est développée dans MILAN, qui représente les différents espaces d'information partagés ou privés existant dans une Salle de Réunion Virtuelle [HAMMA 91].

Enfin, en 1991 au cours du travail du groupe MOCCA du projet européen CO-TECH, Steve Benford, défend la métaphore de ROOM et propose de l'utiliser comme métaphore centrale des environnements de travail coopératif [BENFO 91].

C'est à cette même époque que nous avons fait le choix de la métaphore de ROOM comme métaphore principale de Co-Learn. Notre but est d'utiliser la Salle Virtuelle comme unité élémentaire de structuration de l'espace d'organisation et comme unité macroscopique de l'espace d'activité. Vue de l'espace d'organisation, la Salle Virtuelle est associée à un Travail de Groupe. Vue de l'espace d'Activités, la Salle est un espace d'Activités regroupant des **Personnes**, des **Ressources** et des

Outils. Une personne dans une salle devient un **Participant**, en raison de l'ensemble des **Rôles** qui lui sont confiés dans les différentes Activités de la Salle Virtuelle. Initialement, nous associions à une Salle un type de Situation de Coopération. Les outils présents dans la salle créaient l'environnement favorable au type de situation concernée. Par exemple une audioconférence et une fenêtre partagée sont présents dans les salles dédiées à une situation d'atelier de travail.

Initialement choisie comme une métaphore d'interface, la ROOM deviendra aussi au fil du projet une métaphore de conception puis un objet de première classe dans la réalisation.

5. LA METAPHORE D'OUTIL ET LE PARADIGME D'ACTIVITE

Le choix de la métaphore de l'outil traduit le choix d'un mode d'interaction où l'on ne considère ni l'homme comme une machine, ni la machine comme un humain, mais où on assume la différence entre les deux, en sachant que l'un ne peut pas travailler sans l'autre.

D'autre part, un Outil est une application qui permet de participer à une Activité de groupe. Une Activité est pour nous une Application qui possède un composant de Médiation. L'outil doit donc avoir les propriétés de ce qui est appelée une application coopérante : il doit avoir une interface de communication ouverte à des agents externes (entre autres à un Agent de Rôle).

Il n'est pas facile de décider si un outil doit ou non posséder un Noyau Fonctionnel ou s'il ne doit être qu'un composant Présentation. Ce qui est important, c'est que le Noyau Fonctionnel soit séparé de la Présentation, afin de pouvoir intercaler entre eux un mécanisme de diffusion et de médiation.

CHAPITRE 9

LE CADRE DE L'ANALYSE

Nous présentons dans ce chapitre une compilation des approches d'analyse et de conception qui ont influencé notre conception du système Co-Learn ou ont été utilisées a posteriori pour l'analyser.

La complexité et l'étendue d'un environnement d'aide à la coopération dans une organisation nécessite la mise en oeuvre de nombreuses méthodes d'analyse et de conception considérant les différents aspects du système.

1. LES METHODES D'ANALYSE CONSIDEREES DANS NOTRE ETUDE

1.1. ODP : Analyser les systèmes distribués ouverts

Un environnement CSCW doit d'abord constituer un système distribué ouvert : c'est pourquoi nous utilisons les **cinq points de vue** du modèle ODP [LININ 91] pour structurer notre compilation. Chaque tableau considère un point de vue de ODP et la composante correspondante de chacune des approches considérées.

1.2. MOCCA : Analyser les environnements distribués ouverts de CSCW

MOCCA [BENFO 91] [BENFO 93] [MOCCA 90] [MOCCA 91] [NAVAR 92] propose d'étudier les environnements CSCW. Comme ODP, il utilise plusieurs perspectives comparables. Il propose six points de vue, soit un de plus qu'ODP, parce qu'il partitionne le point de vue de l'entreprise en deux perspectives : l'organisation et l'expertise de l'utilisateur.

Le groupe de travail 2, baptisé MOCCA, du projet européen CO-TECH a identifié des perspectives, pour assister l'analyse, la conception, le développement et l'intégration des environnements socio-techniques d'aide au travail coopératif. Ces perspectives sont maintenant appelées modèles par leurs auteurs, mais elles ne proposent pas de formalisme.

La motivation du travail de MOCCA est qu'il existe maintenant une multitude de logiciels d'aide à la coopération qui s'ignorent. Or chacun de ces logiciels offre des fonctionnalités nécessaires mais insuffisantes pour assister une coopération durable au

sein d'une organisation. L'idée est donc de créer un environnement d'accueil d'applications coopératives, comparable à un système d'exploitation distribué ouvert coopératif.

Les six points de vue proposés s'articulent autour d'une métaphore centrale d'interaction et de conception proposée par Steve Benford [BENFO 91] : la **Salle Virtuelle** (Room). Au coeur du **Modèle des Salles Virtuelles**, appelé aussi **Modèle de l'(Inter-)Activité**, elle est un espace virtuel partagé qui met en présence des **Outils** et des **Personnes** et définit un contexte organisationnel, un espace de travail ou une tâche. Steve Benford dans [MOCCA 92] restreint le rôle des outils au support des **activités asynchrones et structurées**. La Salle Virtuelle enrichit les interactions sociales imprévues et encourage la considération des activités et des personnes. L'interconnexion des Salles Virtuelles forme un espace structuré de gestion et de navigation qui participe au **Modèle Organisationnel**. La Salle Virtuelle délimite un domaine du **Modèle d'Information**. La Salle Virtuelle fournit aux activités qui s'y déroulent les moyens de communication définis par le **Modèle de Communication**. La participation d'un utilisateur aux activités d'une Salle Virtuelle participe au **Modèle de l'Expertise de l'Utilisateur**. Dans le **Modèle de l'Architecture Distribuée**, la Salle Virtuelle définit un domaine de gestion de la localisation des entités de l'environnement (personnes, outils).

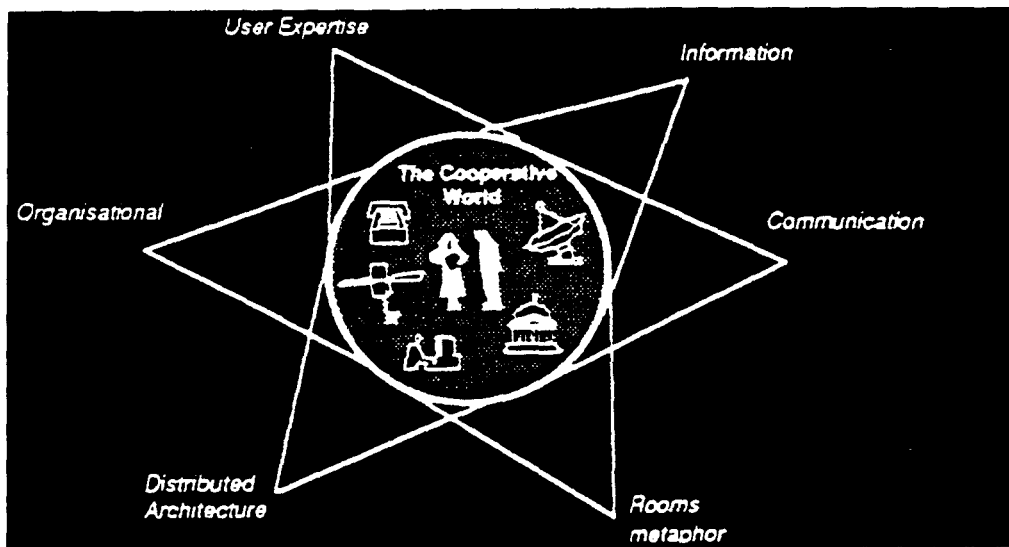


Figure 31. Les six points de vue de MOCCA [MOCCA 90]

1.3. ORDIT : Déterminer les besoins de l'organisation en matière de Système d'Information pour le CSCW

L'approche ORDIT (Object Requirements Definition for Information Technology) [DOBSO 91] propose une méthodologie pour analyser et représenter les besoins des organisations en matière de technologie de l'information et en particulier en matière de systèmes CSCW. Il s'agit de déduire de l'analyse de l'organisation les spécifications fonctionnelles du système technologique mais aussi d'identifier les caractéristiques de l'entreprise qui ne peuvent pas être supportées fonctionnellement par la technologie. Ces données devront être prises en compte par l'organisation pour l'intégration et l'adoption du système dans l'environnement social de l'entreprise.

Uta Pankok-Babatz dans [PANKO 89b, p. 49] constate que les méthodes existantes de description de l'entreprise présentées dans [GROCH 82] ne permettent pas de représenter les activités d'une façon qui permettrait l'automatisation de la coordination nécessaire à leur exécution. Ces méthodes sont l'organigramme, les profils d'emploi, les ordinogrammes, les formulaires et les tables de décision. Chacune de ces techniques de description prend en compte une partie de la description des activités qui se déroulent dans l'entreprise. Mais il n'existe pas de méthode pour décrire l'articulation de ces différentes descriptions, de sorte qu'il est nécessaire de modéliser la même activité de façon redondante par différents moyens. Uta Pankok-Babatz retient en particulier le besoin de modéliser l'articulation des rôles (représentés par les organigrammes et les profils d'emploi), les ordinogrammes et les formulaires. Elle en conclue qu'il faut comprendre les processus de communication de groupe et développer de nouvelles techniques, spécifiques et adaptées à la communication électronique, qui permettent le support informatisé.

Le projet ORDIT du programme ESPRIT II a commencé en 1989 et répond à cette carence. Il développe une méthodologie permettant aux équipes de conception de représenter les buts, les règles et les structures d'une organisation ainsi que les rôles des utilisateurs finaux d'une façon qui facilitera l'identification et l'expression des besoins organisationnels des systèmes technologiques de l'information, et en particulier les systèmes C.S.C.W. pour supporter ces structures et ces rôles. Les membres du projet ORDIT insistent sur leur volonté de permettre la production de systèmes de technologie de l'information qui répondent non seulement aux besoins fonctionnels et organisationnels des utilisateurs individuels mais aussi à ceux des groupes d'utilisateurs avec leurs exigences en matière d'utilisabilité et d'acceptabilité.

Selon ORDIT, l'essence de la coopération est un dialogue effectif entre les rôles à l'intérieur d'une entreprise.

La méthodologie proposée par ORDIT consiste à déterminer dans l'organisation :

- les **Ressources**,
- les **Activités**,
- les **Agences**,
- leurs relations.

Les agences permettent de distinguer les objets sociaux des objets techniques. Les **objets sociaux** ont une agence : ils sont appelés des **Agents**. Un agent ne doit pas être confondu avec la personne qui incarne l'agent : une même personne peut incarner plusieurs agents. Un agent n'est pas non plus un rôle car remplir un rôle peut nécessiter d'incarner plusieurs agents. Une agence est une collection particulière d'obligations.

Les relations entre agents sont définies par des **Rôles Structurels**. Les fonctions qu'un agent assure dans le cadre d'une activité constitue un **Rôle Fonctionnel**. L'activité définit les **Modes d'Accès aux ressources**. Les **Droits d'Accès** d'un agent aux ressources dépendent de son rôle fonctionnel. Les rôles structurels impliquent des rôles fonctionnels : pour remplir ses obligations, donc son rôle structurel, vis-à-vis d'un autre agent, un agent reçoit les moyens et les droits nécessaires pour accéder aux ressources de l'organisation.

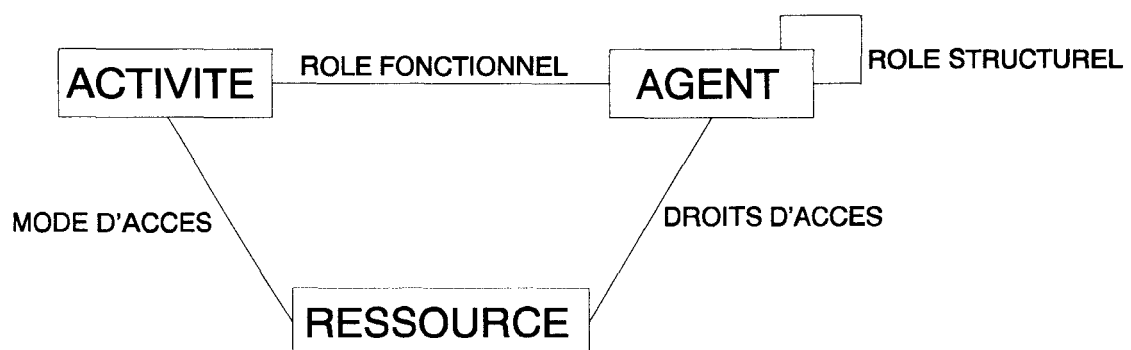


Figure 32. Le modèle ORDIT de l'organisation [DOBSO 91]

ORDIT propose un formalisme graphique pour modéliser les relations entre les activités, les agents et les ressources et mettre en évidence les rôles structuraux et fonctionnels.

Un agent mène une activité au moyen d'**Actes Instrumentaux**, qui agissent sur les ressources, et d'**Actes de Langage**, qui agissent sur les autres agents.

ORDIT propose d'exploiter trois types d'actes de langage : les actes propositionnels (exprimant un fait), les actes illocutoires (exprimant une intention, un

engagement) et les actes perlocutoires (provoquant un changement de sentiment, d'attitude ou de comportement des interlocuteurs). Une **Conversation** est définie comme une séquence d'illocutions qui a pour objet de remplir un but social spécifique. C'est l'unité de conversation la plus large du modèle. Ce qu'ORDIT appelle Speech Act Pattern et que nous traduisons par Echange est une série d'actes de paroles qui forme une unité logique distincte d'une conversation. Les relations entre les actes d'un échange sont définies par un programme de conversation. Les programmes de conversation incluent des **règles procédurales** qui définissent les **actions possibles** à chaque étape de la conversation.

Une fois la modélisation globale de l'organisation réalisée au moyen de la méthode ORDIT, il est possible d'approfondir l'analyse des activités de bureaux. De nombreuses méthodes d'analyse de ces activités ont été développées. La majorité d'entre elles s'intéressait uniquement à la circulation de l'information dans les bureaux. La communication était considérée uniquement sous son aspect transfert d'information. Le lecteur intéressé trouvera une argumentation et un état de l'art de ces méthodes dans [AURAM 92b].

SAMPO [Ibid.] est une approche de modélisation qui est au contraire basée sur une analyse du discours : la communication est considérée comme un moyen d'agir sur les autres. Son analyse du discours repose sur la théorie du langage action et la théorie de la conversation. Les auteurs de SAMPO considèrent classiquement que les actes essentiels du langage action sont les actes illocutoires (affirmation, promesse, ordre, requête, rapport, opinion). Ils définissent un acte illocutoire par trois attributs : le contenu, le contexte et la force illocutoire.

Le contenu est simplement le contenu propositionnel, les mots qui sont prononcés (par exemple : je le ferai demain). La force illocutoire permet de comprendre le but de l'énoncé du contenu (par exemple : est-ce une promesse, une prédiction ou une boutade ?) .

Le contexte permet de comprendre le non-dit. Par exemple, la connaissance de ce qui a été dit avant, l'identité et le rôle des personnes à qui le message est destiné, le lieu où la conversation se déroule permet de savoir quelle tâche représente le pronom 'le' dans le contenu propositionnel. La date à laquelle la phrase a été prononcée permet de savoir quel jour représente 'demain'. L'identité de celui qui a prononcé ou écrit le message renseigne sur qui est 'je'. SAMPO définit une conversation comme un type d'échange cohérent et complet qui supporte l'accomplissement des tâches du bureau. Les auteurs insistent sur l'importance de la nature systématique, similaire à un jeu, des

conversations : chaque acte illocutoire crée la possibilité d'un ensemble fini et habituellement assez limité d'actes illocutoires utilisables pour répondre.

SAMPO considère que les actes instrumentaux peuvent être représentés par des actes perlocutoires. SAMPO présente l'intérêt majeur de permettre de représenter la connexion entre les actes de langages et les actes instrumentaux. Le lecteur pourra trouver une comparaison intéressante de la méthode SAMPO avec les méthodes O.S.S.A.D. et I.C.N. dans [AURAM 92a]. L'utilisation de SAMPO pour analyser les activités de l'organisation constitue une étape intéressante entre la modélisation générale de l'organisation telle qu'elle peut être réalisée à l'aide de la méthode ORDIT, et une modélisation très fine des activités de groupe telle qu'elle peut être obtenue en utilisant une méthode dérivée de AMIGO.

ORDIT intervient dans trois points de vue ODP : les points de vue de l'entreprise, de l'information et de la dynamique.

1.4. AMIGO : Concevoir les systèmes de coordination des activités de communication de groupe

AMIGO [PANKO 89] [SMITH 89] permet de concevoir les activités de communication de groupe.

A la suite d'une analyse de l'organisation menée à l'aide de la méthodologie ORDIT, on peut utiliser la méthode développée par le groupe AMIGO pour approfondir l'analyse des activités de communication asynchrone de groupe.

Le groupe AMIGO identifie quatre types de modèles d'analyse des activités de communication de groupe (ne retenant que les activités de communication en temps différé) :

- **les modèles orientés procédures,**
 - ils se concentrent sur la modélisation d'une application spécifique,
 - ils sont plutôt adaptés à une architecture centralisée,
 - ils sont bien adaptés aux activités concurrentes,
 - ils utilisent principalement les réseaux de Petri pour décrire les schémas de coordination
- **les modèles orientés formulaire,**
 - ils se concentrent sur la modélisation d'une application spécifique,
 - ils reposent sur le concept de messages structurés et intelligents,
 - ils sont bien adaptés à des activités séquentielles,
 - ils utilisent principalement les Règles pour décrire les schémas de coordination,

- **les modèles orientés structure de la communication,**
 - ils sont indépendants de l'application,
 - ils se concentrent sur la modélisation des structures personnelles et organisationnelles qui fondent les structures de communication d'un groupe,
 - ils introduisent le concept de chemin,
- **les modèles orientés conversation,**
 - ils sont indépendants de l'application,
 - ils sont fondés sur la théorie de l'acte de langage,
 - les schémas de communication sont définis à l'avance et immuables.

Le modèle d'activité AMIGO est plutôt orienté procédure. Cependant, il en diffère par la souplesse supplémentaire qu'il offre en permettant de définir le cadre et les contraintes au lieu de décrire une régulation stricte des procédures de communication de groupe. Pour ce faire, AMIGO utilise une technique similaire à des règles de production.

Le modèle d'activité AMIGO diffère de l'approche orientée formulaire parce que les Règles ne sont pas associées aux messages mais sont connues des Agents dès le début de l'Activité. Cependant, comme l'approche formulaire, AMIGO utilise des messages typés structurés.

L'approche d'AMIGO est différente de celles des modèles orientés communication et conversation dans la mesure où ces modèles visent à supporter les activités de conversation, alors qu'AMIGO est conçu pour supporter les activités instrumentales.

Wolfgang Prinz [PRINZ 89, p. 178] note qu'il pourrait être intéressant de combiner les approches communication et conversation avec le modèle AMIGO pour fournir à l'utilisateur un support de communication compréhensif pour supporter ses activités de communication en relation avec ses activités instrumentales. Nous verrons plus loin que nous avons réalisé cette combinaison d'AMIGO et des approches communication et conversation pour analyser et concevoir à la fois les activités instrumentales et les activités de conversation dans le cadre de Co-Learn.

AMIGO définit un **Modèle d'Activité** par des **Rôles**, des **Règles**, des **Messages** et des **Fonctions**. Une Activité est menée par un ensemble d'Agents.

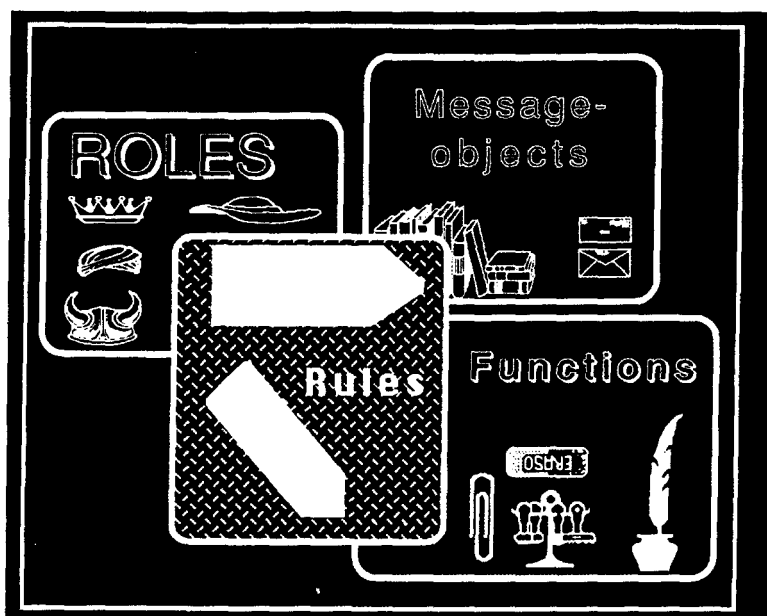


Figure 33. Le modèle AMIGO, extrait de [DANIE 89]

Un Rôle définit l'ensemble des Types de Messages que l'Agent peut recevoir et l'ensemble des Types de Messages que l'Agent peut envoyer, et un ensemble d'attributs qui définissent l'Etat du Rôle.

Les Règles définissent des règles d'évolution de l'Activité qui sont activées en réaction à l'occurrence d'un événement prévu (par exemple, l'émission d'un Type de Message par un Agent). Ces Règles testent un certain nombre de conditions liées à l'état global de l'activité, à la valeur des attributs du message émis, au rôle et à l'état de l'émetteur et déclenchent les Fonctions adéquates. Ces fonctions peuvent être des **Fonctions Génériques** (par exemple, envoyer un message aux membres du groupe jouant un certain rôle, modifier le stade de l'activité, modifier le rôle de certains membres) ou des **Fonctions Spécifiques** à l'activité (par exemple, pour une activité de vote, mettre à jour les résultats).

Le lecteur trouvera une description détaillée de la méthode AMIGO dans [AMIGO 89].

1.5. [ELLIS 94] : Permettre aux utilisateurs d'analyser les collecticiels

Le modèle conceptuel proposé par Ellis lors de la conférence CSCW'94 est constitué de trois modèles complémentaires. Le **Modèle de l'Interface Utilisateur** s'intéresse aux vues adaptées que le système informatique offre aux utilisateurs pour améliorer leur interaction : vues sur les objets d'information partagés, vues sur les participants, vues sur le contexte de la coopération. Il appartient au modèle de l'organisation. Le **Modèle Ontologique** définit les objets et les opérations que le système offre aux utilisateurs. Il contribue à analyser le point de vue de l'information

ODP. Le **Modèle de la Coordination** analyse la coordination des activités menées par les participants. Il participe à compléter le point de vue de l'organisation et n'est pas sans rappeler le modèle ORDIT.

1.6. [HENNE 94] : Intégrer les collecticiels par le partage d'un modèle commun de l'entreprise

[HENNE 94] propose un métamodèle objet pour représenter l'information sur l'entreprise. Nous nous intéressons ici à l'analyse qui précède la proposition de ce métamodèle. Deux types d'éléments du modèle de l'entreprise sont identifiés : les objets et les concepts de l'entreprise. Les objets de l'entreprise sont les **Personnes**, les **Ressources** et les **Lieux**. Les concepts sont les **Rôles**, les **Groupes**, les **Règles**, les **Relations** et les **Structures**.

Le but est d'offrir aux utilisateurs et aux applications une base d'information commune sur l'entreprise pour situer les activités dans leur contexte organisationnel.

2. ARTICULATION DES APPROCHES

2.1. Contributions au point de vue de l'ORGANISATION

APPROCHE & COMPOSANTE	DONNEES CONSIDEREES	MODELES UTILISES	OBJECTIFS
ODP POINT DE VUE DE L'ENTREPRISE	Ce sont les informations à propos de l'organisation [LININ 91] Politiques de gestion et de sécurité de l'entreprise Rôles dans l'entreprise Contraintes de l'entreprise Objectifs de l'entreprise Activités dans l'entreprise Actions possibles dans l'entreprise Rôles dans les activités dans l'entreprise Obligations dans l'entreprise Situations dans l'entreprise [GRIET 91] Structure de l'entreprise Infrastructure d'information de l'entreprise Interaction utilisateurs/système Affaires, Situations, Comportements dont l'entreprise doit être informée Rôles des personnes nécessitant un SI Activités des personnes nécessitant un SI Moyens de traitement de l'information Localisation des moyens de traitement Politique de protection et de gestion de l'information	Modèles d'Organisation (organigrammes) Modèles d'Activité (ordinogrammes, diagrammes de flux, schémas fonctionnels) Exemple : SADT Modèles sémantiques (extensions du modèle relationnel) Exemple : INFOMOD Modèle structurel Relations Cardinalités Entités Types d'entités Attributs Domaines des attributs Règles sémantiques Règles d'identification Règles d'interdépendance Règles de dérivation Règles de classification Règles de sous-typage Règles de transition d'état Spécification de l'échange d'information Squelettes des messages Spécification d'action Règles d'autorisation d'action Vues, sous-modèles	Prendre en compte les besoins des utilisateurs Offrir une vue de l'utilisation du système aux utilisateurs Offrir un support de discussion avec les utilisateurs

<p>MOCCA <i>POINT DE VUE DE L'ORGANISATION</i></p>	<p>Objets organisationnels Employés Coordonnées Comités Départements Salles Types de documents Relations organisationnelles Rôles Règles Responsabilités Contrôle d'accès Nommage</p>	<p>AOO : Classes Objets Attributs</p>	<p>Concevoir une base de connaissance organisationnelle qui puisse : Fournir des informations sur l'organisation aux utilisateurs Fournir un contexte commun aux applications Permettre l'identification des objets de l'organisation Interagir avec X500</p>
<p>MOCCA <i>MODELE CONCEPTUEL DES SALLES VIRTUELLES</i></p>	<p>Espace d'interaction virtuel partagé, Contexte de travail coopératif, Réunion d'outils, Réunion de participants, Coprésence, Intégration d'activités,</p>	<p>Inexistant</p>	<p>Décrire et caractériser le travail coopératif Permettre en permanence la communication informelle entre les utilisateurs Donner conscience de la présence et de l'activité des autres participants Contextualiser socialement l'utilisation des outils</p>
<p>MOCCA <i>POINT DE VUE DE L'EXPERTISE UTILISATEUR</i></p>	<p>Toute information pouvant améliorer l'interaction entre les utilisateurs Utilisateur : expérience expertise préférences rôles tâches en cours</p>	<p>Inexistant</p>	<p>Offrir au système ou au concepteur les données nécessaires pour adapter les interfaces aux utilisateurs Intégrer au système la connaissance de l'organisation et de ses membres afin qu'il s'y adapte en permanence Offrir aux utilisateurs les informations utiles au choix de leurs collaborateurs</p>

<p>ORDIT <i>MODELE DE L'ACTIVITE</i></p>	<p>Objectifs de l'entreprise Politique de l'entreprise Contraintes de l'entreprise Rôles Affectation des rôles Décideurs</p>	<p>Enterprise Model Entités organisationnelles Activités Agents Ressources Relations organisationnelles Rôles structurels Rôles fonctionnels Droits d'accès Modes d'accès Actes instrumentaux Actes de parole Conversations Role Reference Model</p>	<p>Améliorer la communication entre les concepteurs et les utilisateurs Déduire les spécifications fonctionnelles du système technologique d'information Identifier les enjeux non fonctionnels qui ne peuvent être traduits en terme de spécifications du système</p>
<p>Ellis 94 <i>MODELE DE L'INTERFACE UTILISATEUR</i></p>	<p>Vues offertes par le système à l'utilisateur : sur les objets : objets d'information méta-objets opérations locales, télé pointeurs, fenêtres partagées sur les Participants : Téléprésence, aura, nimbus, focus, informations sociales, expérience, profil, situation géographique sur le Contexte : structurel social : normes, historique, métriques du groupe, dramaturgie organisationnel : hiérarchie, règles, statuts.</p>	<p>Inexistant</p>	<p>Permettre de caractériser les collecticiels pour les décrire et les comparer</p>

Ellis 94 <i>MODELE DE LA COORDINATION</i>	Activités des participants, Tâche = instance d'activité, Opérations sur les objets, Accès partagé aux objets, Coordination des activités, Procédures, Processus (Endeavour) = instance de procédure Actions collaboratives, Rôles, Acteurs, But, Etape (Phase), Concurrence des tâches, La coordination au niveau activité simplifie la coordination au niveau objet	Inexistant	Permettre de caractériser les collecticiels pour les décrire et les comparer
AMIGO AAM <i>CONTEXTE ORGANISATIONNEL DE LA COMMUNICATION DE GROUPE</i> (EXTENSION : P. 303)	Types de messages, Relations entre types de messages et : - les rôles, - les conversations, - les structures organisationnelles, - les activités	Inexistant	Fournir un contexte aux activités de communication de groupe
Hennessy 94 <i>MODELE DE L'ENTREPRISE</i>	Objets Personnes, Ressources, Lieux Concepts : Procédures Tâches Activités Rôles, Groupes, Règles, Relations, Structures	Inexistant	Fournir un contexte de travail cohérent
OOA <i>DOMAINES DE L'APPLICATION</i>	Système désiré par l'utilisateur final	Inexistant	

Table 5. Les points de vue sur l'organisation

2.2. Contribution aux points de vue de l'INFORMATION

APPROCHE & COMPOSANTE	DONNEES CONSIDEREES	MODELES	OBJECTIFS
ODP <i>POINT DE VUE DE L'INFORMATION</i>	L'information manipulée dans l'entreprise Structures des éléments d'information Attributs de qualité des éléments d'information Flux d'information Règles régissant les relations entre les éléments d'information Forme de présentation de l'information et des traitements aux utilisateurs Changements subis par les éléments d'information	Modèles d'information	
MOCCA <i>MODELE DE L'INFORMATION</i>	Toute information produite, consommée et partagée au cours de la communication Messages Documents Conversations Relations Domaines (Partitions), Spécifications des S.G.B.D., Vues Spécification du contrôle de transaction Gestion de schéma Contrôle d'accès	Modèle orienté objet : Objets Méthodes Hiérarchie ... GCS [SMITH 91] [BENFO 92]	Permettre le partage de l'information entre les applications en permettant de structurer leurs informations dans un format commun et en offrant un schéma de nommage global
ORDIT <i>SCHEMA CONCEPTUEL</i>	Ressources Relations entre les ressource Opérations sur les ressources	Modèle graphique ORDIT	
GCS <i>MODELE D'INFORMATION</i>	Entités Liens Domaines Items		

Ellis 94 MODELE ONTOLOGIQUE	Classes d'objets manipulés par l'utilisateur : attributs, valeurs Instances d'objets Relations entre objets Opérations offertes à l'utilisateur : de consultation, de création, de modification, de destruction Actions sur les objets du travail Actions sur le groupe et l'environnement Sémantique intentionnelle Sémantique opérationnelle Droits d'opération, Droits d'accès, Propriété, protection, contrôle Valeurs automatiques	Modèles d'information (Modèle Entité-Relation, Modèle Objet)	Offrir une description statique des objets manipulés par les utilisateurs
OMT MODELE OBJET (ANALYSE)	Classes Objets Relations : agrégation, spécialisation, généralisation, association Attributs Opérations	Diagrammes de structure	
OMT CONCEPTION OBJET (CONCEPTION)	Adaptation aux contraintes du système		
OOA INFORMATION (ANALYSE)	Domaines, Classes (Entités), Attributs, Clés, Relations, Attribut référentiel, Héritage	Diagrammes de structure	
OOA DOMAINES DE SERVICE (CONCEPTION)	Mécanismes génériques, Fonctions utilitaires		

<p>AMIGO MHS+ <i>INFORMATION</i></p>	<p>Objets échangés ou manipulés au cours d'un processus de communication de groupe Types d'objet d'information messages documents conversations Objets atomiques Attributs typés : de nommage statiques dynamiques Partage des objets d'information Identité des objets d'information Structure des objets d'information Opérations sur les objets : lire modifier ajouter supprimer Nommage des objets d'information Nommage des entités communicantes Environnement d'interaction Droits et protocoles d'accès Groupes d'entités de communication Ensembles d'objets d'information Composition d'objets d'information Règles de dérivation</p>	<p>X500</p>	
<p>ODA</p>	<p>Documents Structures Identifiants</p>		
<p>X500 <i>SERVICE DE</i> <i>REPertoire</i></p>	<p>Classes d'objets Attributs</p>		

Table 6. Les points de vue sur l'information

2.3. Contribution aux points de vue DYNAMIQUES

APPROCHE & COMPOSANTE	DONNEES CONSIDEREES	MODELES	OBJECTIFS
ODP <i>MODELE LOGICIEL</i>	Interaction client-serveur: invocation publication Interfaces des objets Instanciations Réseaux d'objets Identification de fonctions génériques Construction : règles d'association Communication Synchronisation Applications Intégration d'applications	Types Abstraits de Données	
MOCCA <i>MODELE DE LA COMMUNICATION</i>	Primitives d'aide à la communication Canaux de communication Structure : des messages échangés, des événements Relations entre : les événements, les messages, les activités Intégration d'activités		Offrir aux applications de communication intégrées dans l'environnement la possibilité d'interagir entre elles et avec l'environnement
AMIGO (AAM) <i>MODELE D'ACTIVITE</i>	Communicants, Actes, Messages, Rôles, Règles, Fonctions		Analyser et concevoir le support des activités de communication humaine en temps différé
AMIGO MHS+ <i>MODELE D'INTERACTION</i>	Interaction entre les entités communicantes et les objets, Environnements, Entités applicatives,		
OMT <i>MODELE DYNAMIQUE (ANALYSE)</i>	Etats, Transitions, Evénements, Séquences d'interaction, Simulation (Mock-up)		
OMT <i>MODELE FONCTIONNEL (ANALYSE)</i>	Processus, Flux de données, Contraintes entre objets : préconditions, post-conditions		

<p>OOA <i>MODELE D'ETAT</i> (ANALYSE)</p>	<p>Etats, Transitions, Actions, Modèle de communication objet</p>		
<p>OOA <i>MODELE DE</i> <i>PROCESSUS</i> (ANALYSE)</p>	<p>Processus : - d'accès aux données, - générateurs d'événements, - de transformation, - de test Mémoire de donnée objet. Flux de données. Conditions sur les flux. Modèle d'accès aux objets, Table processus/état</p>		

Table 7. Les points de vue dynamiques

2.4. Contribution aux points de vue de l'INGENIERIE

APPROCHE & COMPOSANTE	DONNEES CONSIDEREES	MODELES	OBJECTIFS
ODP <i>MODELE DE L'INGENIERIE</i> MOCCA (?)	Mécanismes de contrôle, Edition de liens, Edition automatique de source; Mécanismes de transparence, Processeurs, Mémoires, Réseaux de communication, Choix de mécanismes inter- processus Formats de média riches, Protocoles de communication + réactifs, Plates-formes de distribution plus adaptables, Extension du modèle client- serveur à des modèles avec plus de partage X400 MHS, X500 ODA EDI		Optimiser la réalisation du modèle logiciel
OMT <i>CONCEPTION SYSTEME (CONCEPTION)</i>	Couches, Partitions, Tâches concurrentes, Communication Inter- processus Allocation des ressources, Caractéristiques matérielles, Choix des canaux de communication, Choix des protocoles de communication, Choix des systèmes de stockage : Système de Gestion de Base de Données, Système de Gestion de Fichiers Choix des techniques de contrôle logiciel : procédural, événementiel, concurrentiel		
AMIGO (AAM) <i>ARCHITECTURE</i>	Distribution Connexion à X400/X500 Services : Archivage, Distribution, Répertoire		

OOA DOMAINES ARCHITECTURAUX (CONCEPTION)			
---	--	--	--

Table 8. Les points de l'ingénierie

2.5. Contribution aux points de vue de la TECHNOLOGIE

APPROCHE & COMPOSANTE	DONNEES CONSIDEREES	MODELES	OBJECTIFS
ODP MODELE TECHNOLOGIQUE	Composants techniques du système distribué Matériel Logiciel de base : systèmes d'exploitation, périphériques d'entrée/sortie, points d'accès aux communications		

Table 9. Les points de vue de la technologie

2.6. Influence de MOCCA sur ODP

MOCCA considère que les environnements CSCW sont des systèmes distribués ouverts et que par conséquent les principes de modélisation ODP leur sont applicables.

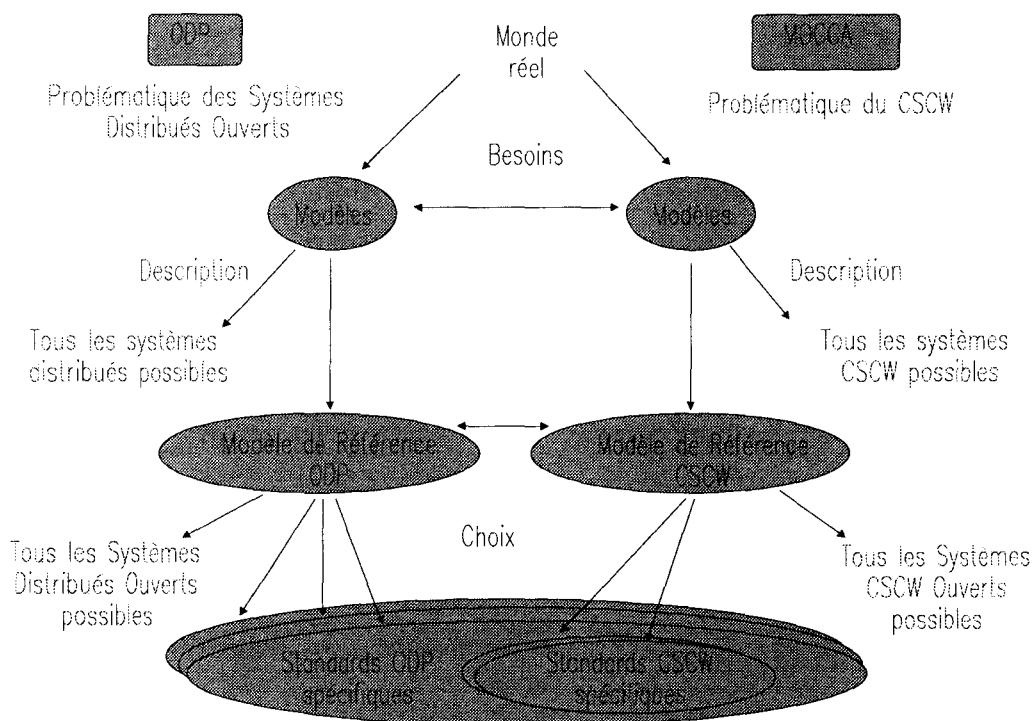


Figure 34. Articulation de MOCCA et ODP, extrait et traduit de [MOCCA 92]

De plus, [MOCCA 92] retient plusieurs effets prévisibles du CSCW sur les points de vue ODP.

2.6.1. Influence sur les points de vue de l'Information et de l'Entreprise

- Alors que la modélisation actuelle de l'entreprise de ODP est descriptive et passive, les modèles de l'entreprise deviendront plus actifs,
- Un flou apparaît entre l'Information et l'Entreprise, dans la mesure où le modèle de l'organisation doit être intégré dans le système d'information. Cette interpénétration des modèles de l'organisation et de l'information est caractéristique des environnements d'aide au travail coopératif : les données sur l'organisation font partie de l'information manipulée par les collaborateurs.

2.6.2. Influence sur le Modèle Logiciel

- Les mécanismes de transparence devraient pouvoir être ajustés par les utilisateurs : la localisation des entités est parfois nécessaire pour coopérer. Cette idée de MOCCA est à rapprocher du besoin de rétroaction de groupe.

2.6.3. Influence sur le Point de Vue de l'Ingénierie

- Des formes de média plus riches sont nécessaires,
- Les protocoles de communication doivent être plus réactifs,
- Les standards pour la communication de groupe et l'échange d'information doivent se développer,
- Le modèle client-serveur doit être étendu vers des modèles plus symétriques. Cette conséquence a déjà été identifiée dans la première partie de cette thèse.

2.7. Articulation de MOCCA, ORDIT et AMIGO

Nous considérons que l'approche ORDIT est bien adaptée pour l'analyse des activités à l'échelle globale de l'entreprise mais que son but n'est pas de supporter la coopération au sein des groupes de travail. Les activités considérées par SAMPO sont typiquement des activités administratives telles que les procédures de prise de commande ou d'organisation de déplacements. Les activités intégrées par MOCCA dans les Salles Virtuelles (par exemple des salles de cours, des salles de réunion, des laboratoires, des salles de projet) ne sont pas de la même nature. Les activités considérées par ORDIT mettent en jeu des relations interpersonnelles (même si les interlocuteurs peuvent être des services) et non les activités d'un groupe au sens où nous l'avons défini dans la première partie de cette thèse. La méthode SAMPO utilisée par ORDIT pourrait servir à modéliser la communication entre les groupes. AMIGO permet d'analyser et concevoir les activités de communication asynchrones de groupe intégrables dans une Salle Virtuelle.

Selon MOCCA, l'analyse de la sémantique des activités ne fait pas partie de la modélisation de l'environnement de CSCW : ce sont des applications spécifiques qui doivent pouvoir être facilement intégrées dans l'environnement, moyennant le développement d'adaptateurs spécifiques. Du point de vue de l'environnement MOCCA, AMIGO permet de développer des applications qui sont vues comme des activités produisant des messages et des événements qu'elles peuvent publier pour être utilisés par les autres applications. AMIGO propose d'analyser le fonctionnement interne de chaque activité isolée et MOCCA propose de concevoir un environnement permettant de faire communiquer un ensemble d'activités de groupe.

3. NOTRE PROPOSITION : UN ENVIRONNEMENT DE CONCEPTION, DE DEVELOPPEMENT ET D'INTEGRATION D'ACTIVITES DE COOPERATION

Nous adoptons une démarche un peu différente de celle du groupe MOCCA : en plus d'un cadre d'accueil d'activités de coopération, nous proposons une méthode de conception et un environnement objet de développement de ces activités de groupe.

Notre environnement d'intégration des activités est fondé sur la métaphore de Salle Virtuelle et le concept d'Outil, comme dans MOCCA.

Notre environnement de développement d'activités de groupe contient des classes génériques que le développeur peut dériver pour créer de nouvelles classes d'activités, et un ensemble de classes d'activités de base, qui illustrent les possibilités du système en matière de développement et d'intégration d'activités coopératives. La méthode que nous utilisons pour concevoir et réaliser ces activités est inspirée de la méthode AMIGO, dans la mesure où elle identifie des entités communicantes qui participent à une activité commune qui est définie par un composant Message, un composant Rôle, un composant Règle et un composant Fonction. Nous l'adaptions¹ de sorte que notre méthode permet de modéliser et d'intégrer uniformément les activités de coopération en temps réel et les activités de coopération en temps différé. Ceci est réalisé entre autres grâce à l'ajout d'un composant Droits, l'extension des Fonctions Génériques et la définition de Messages génériques, qui permettent la visualisation en temps réel de l'évolution des droits des participants et la prise en compte de la rétroaction de groupe. Notre méthode de conception concilie deux approches de la modélisation des activités : l'approche orientée médiateur illustrée par Officetalk-D et DOMINO [PRINZ 89], et l'approche orientée rôle adoptée par les modèles SCOOP et COSMOS [PRINZ 89][BOWER 88].

Par ailleurs, nous augmentons la souplesse de la gestion de la coordination en permettant la définition et la modification en temps réel de plusieurs Modes de Médiation d'une activité, grâce à l'utilisation de messages génériques de modification des droits. Nous ajoutons ainsi la dynamique des modèles orientés formulaires en définissant des types de messages qui transportent des règles. Notre méthode est donc un compromis entre l'approche procédurale de AMIGO et l'approche par les formulaires.

L'environnement que nous proposons est décrit en détail dans son état actuel dans les cinq chapitres qui suivent.

¹Cf. chapitre 10, § 1.2

CHAPITRE 10**LES POINTS DE VUE DE L'ORGANISATION**

Ce chapitre propose l'analyse de Co-Learn sous le point de vue de l'organisation. Le rôle du système Co-Learn est de supporter les situations d'apprentissage coopératif. C'est pourquoi cette analyse commence par le bas de la pyramide (figure 35) et s'intéresse d'abord aux situations d'apprentissages, c'est à dire aux situations de coopération au sein des équipes. Cette approche favorise la prise en compte des utilisateurs et de leurs activités, contrairement à une approche plus classique de l'organisation qui s'intéresserait avant tout à la structure hiérarchique de l'organisation, partant du haut de la pyramide.

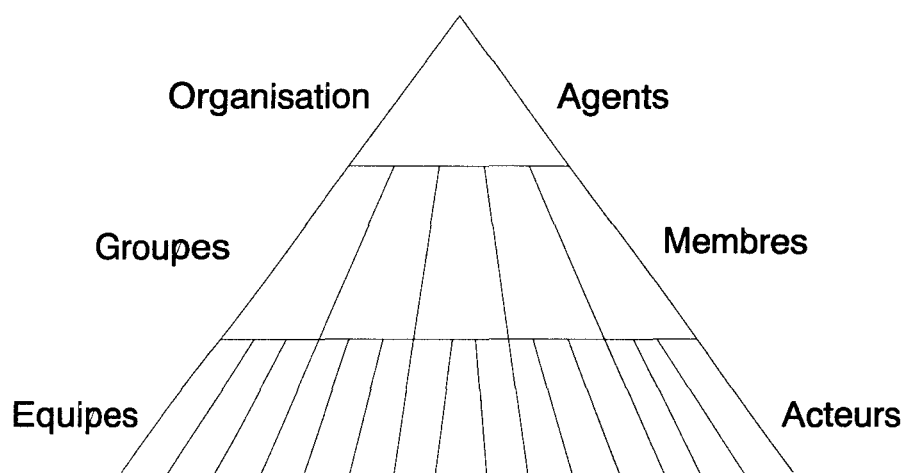


Figure 35. La pyramide de l'organisation

Après avoir analysé et modélisé ces situations de coopération, nous analysons le deuxième étage de la pyramide qui identifie des unités organisationnelles utilisées pour structurer l'univers des situations de coopération du niveau inférieur. Les unités organisationnelles servent surtout à contextualiser les situations de coopération.

1. LA BASE DE LA PYRAMIDE : LES SITUATIONS D'APPRENTISSAGE COOPERATIF

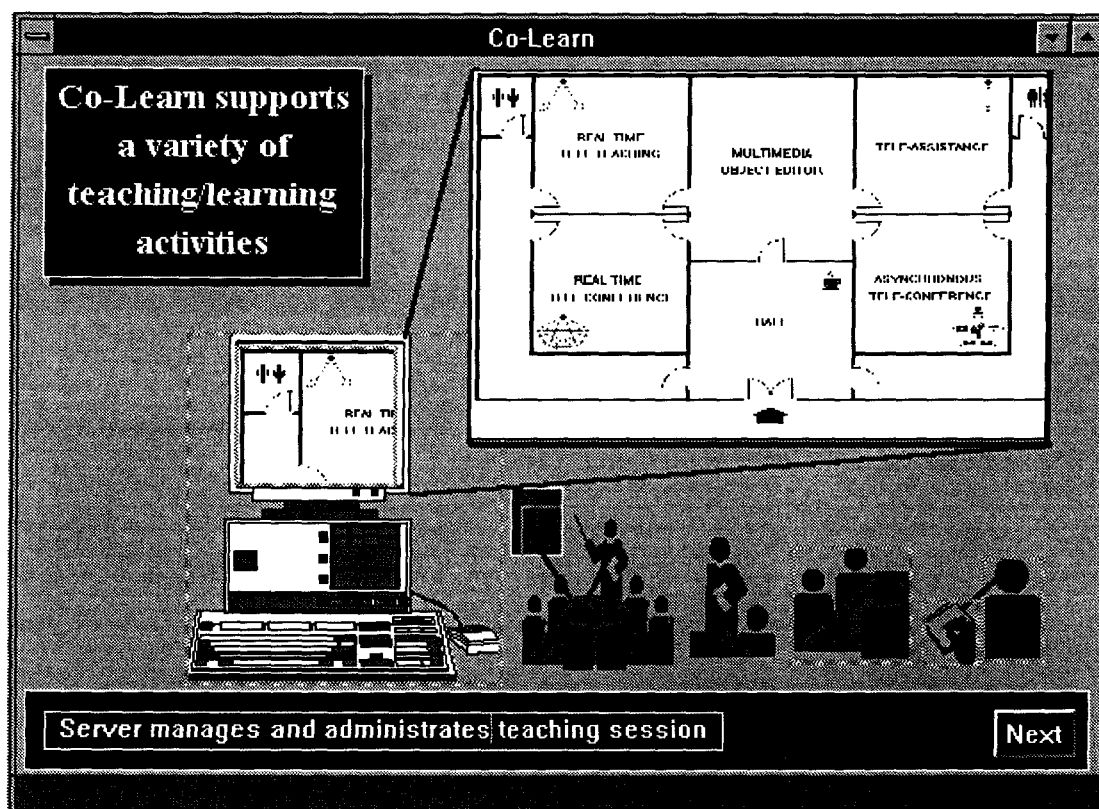


Figure 36. Supporter divers environnements de coopération [SMITH 94]

1.1. Composer des environnements de coopération

Nous ne pouvons pas spécifier toutes les situations d'apprentissage coopératif. C'est pourquoi l'objectif majeur de Co-Learn est la malléabilité : le système Co-Learn doit pouvoir être adapté à l'organisation de formation. Au sein d'une même organisation, les formateurs peuvent utiliser des modes d'apprentissage différents selon la matière à enseigner ou le public. Nous avons donc choisi d'offrir aux formateurs la possibilité de composer des environnements d'apprentissage adaptés aux styles d'enseignement qu'ils désirent pratiquer. Pour matérialiser cette notion d'environnement d'apprentissage aux yeux du formateur, nous utilisons la métaphore de Salle Virtuelle. Les formateurs peuvent créer, configurer, ouvrir, fermer, supprimer autant de salles virtuelles qu'ils le désirent.

1.1.1. Concevoir les situations de coopération

Pour modéliser les situations de coopération, nous nous inspirons de la méthode ORDIT qui identifie trois types d'entité : l'Activité, l'Agent et la Ressource.

Dans notre modèle, les ressources principales sont les Outils¹ de la Salle Virtuelle, puisqu'ils sont offerts aux personnes pour participer aux Activités de l'équipe. Les agents sont les participants inscrits dans la Salle Virtuelle.

Les relations entre Agents et Activités sont définis par les rôles fonctionnels. La relation entre deux agents est généralement médiatisée par une ressource, c'est à dire, dans le cas de Co-Learn, par un outil : la différence de rôle fonctionnel des agents dans une même activité traduit les rôles structurels entre les agents.

Le rôle fonctionnel est opérationnalisé par des droits d'accès aux fonctions de l'outil qui supporte l'activité. L'articulation des rôles fonctionnels autour de l'activité et les droits d'accès à l'outil sont analysés à l'aide d'un modèle dérivé du modèle d'activité de groupe d'AMIGO [DANIE 89].

La relation entre les activités est souvent médiatisée par les outils.

La figure 37 modélise l'organisation des rôles et des ressources impliquées dans les sessions d'apprentissage supportées par Co-Learn.

¹Cette approche des ressources par les outils permet de gérer la coordination au niveau des activités, au lieu de la gérer uniquement au niveau des objets manipulés. Cette approche, que nous avons adoptée au démarrage du projet Co-Learn, en 1992, a été identifiée récemment comme une piste intéressante par [ELLIS 94]

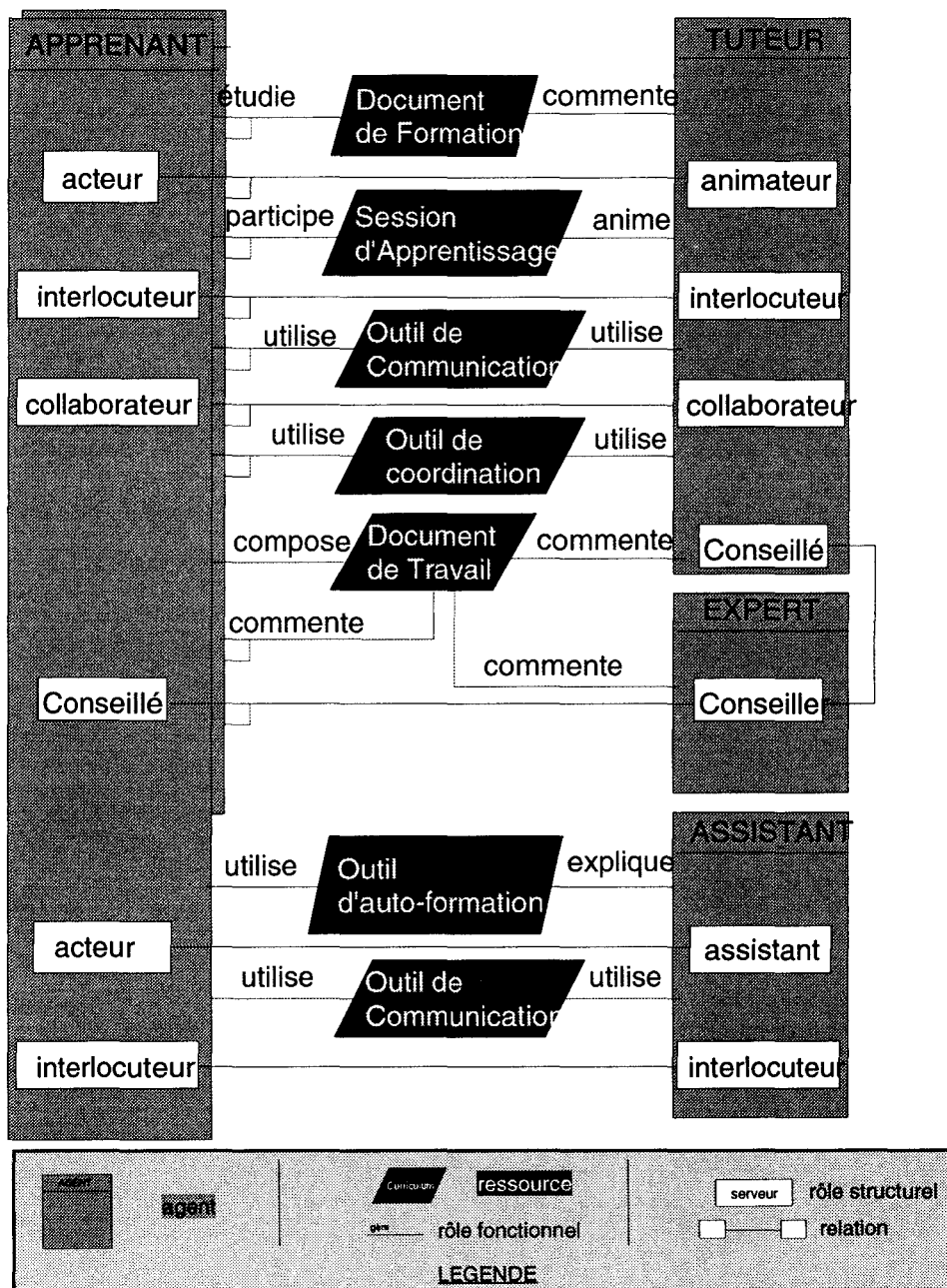


Figure 37. Modélisation de type ORDIT de l'organisation des situations d'apprentissage dans un institut d'enseignement ouvert

Les ressources Document de Formation, Document de Travail, Session d'Apprentissage, Outil de Communication sont partagés par un groupe d'apprenants et un tuteur. Un apprenant utilisant seul un Outil d'auto-formation peut partager un outil de communication avec un assistant qui l'aide dans l'utilisation de l'outil. Le tuteur est un pédagogue, l'expert est un spécialiste d'un domaine (chercheur, industriel, consultant...), l'assistant est un expert de l'utilisation des didacticiels.

1.1.2. Créer une Salle Virtuelle

Un formateur a plusieurs moyens de créer une Salle Virtuelle adaptée à ses besoins :

- instancier un type pré-configuré de Salle Virtuelle,
- instancier un type pré-configuré de Salle Virtuelle et modifier la configuration de l'instance obtenue,
- définir et instancier un nouveau type de Salle Virtuelle soit en le définissant entièrement, soit en modifiant un type existant.

Le système propose quatre types de Salles Virtuelles pré-définies :

- La Salle Virtuelle de télé-assistance [COLEARN 93d] correspond à une composition de la salle de permanence d'Enseignement A Distance, de la salle de Travaux Pratiques et du centre de ressources. Le formateur qui assure la permanence peut utiliser un outil de contrôle de poste distant pour observer et assister le travail individuel des apprenants présents dans la Salle Virtuelle.

- La Salle Virtuelle de télé-enseignement [COLEARN 94a] : correspond à la salle de cours magistral (Amphithéâtre). Elle permet à un enseignant de dispenser un cours magistral à plusieurs grands groupes d'apprenants. Elle opérationnalise donc un mode d'enseignement traditionnel permettant de transmettre simultanément un ensemble d'informations à un large public relativement passif.

- La Salle Virtuelle d'atelier de travail [COLEARN 93f] [CROIS 95] permet à une petite équipe de s'auto-organiser et de coopérer en temps réel. Elle correspond à la situation générale d'une réunion. Ce type de Salle Virtuelle peut accueillir des situations de coopération très variées car il permet de créer des instances qui peuvent elles-mêmes être configurées. En ceci ce type de Salle Virtuelle est très différent de ceux décrits précédemment (Télé assistance et télé-enseignement) pour lesquelles la configuration de l'environnement de coopération est complètement défini par le type et sera donc la même pour toutes les instances.

Ceci est dû à ce que dans toutes les situations de télé-assistance et de télé-enseignement, les rôles des participants et donc le type de relation qui existe entre eux sont constants. D'abord, il n'y a qu'un type de relation dans les deux premières situations : celle entre l'enseignant et l'apprenant. Les apprenants n'ont aucune relation entre eux. On se ramène toujours à la situation d'une paire.

Par contre, dans une situation de réunion, les relations peuvent varier car tous les participants peuvent être en relation les uns avec les autres. On ne peut pas se ramener à des relations interpersonnelles : on est face à un vrai groupe.

Etant donnée la richesse potentielle de la coopération dans une réunion, la Salle Virtuelle d'atelier de travail peut utiliser beaucoup d'outils et mettre en jeu de nombreux rôles.

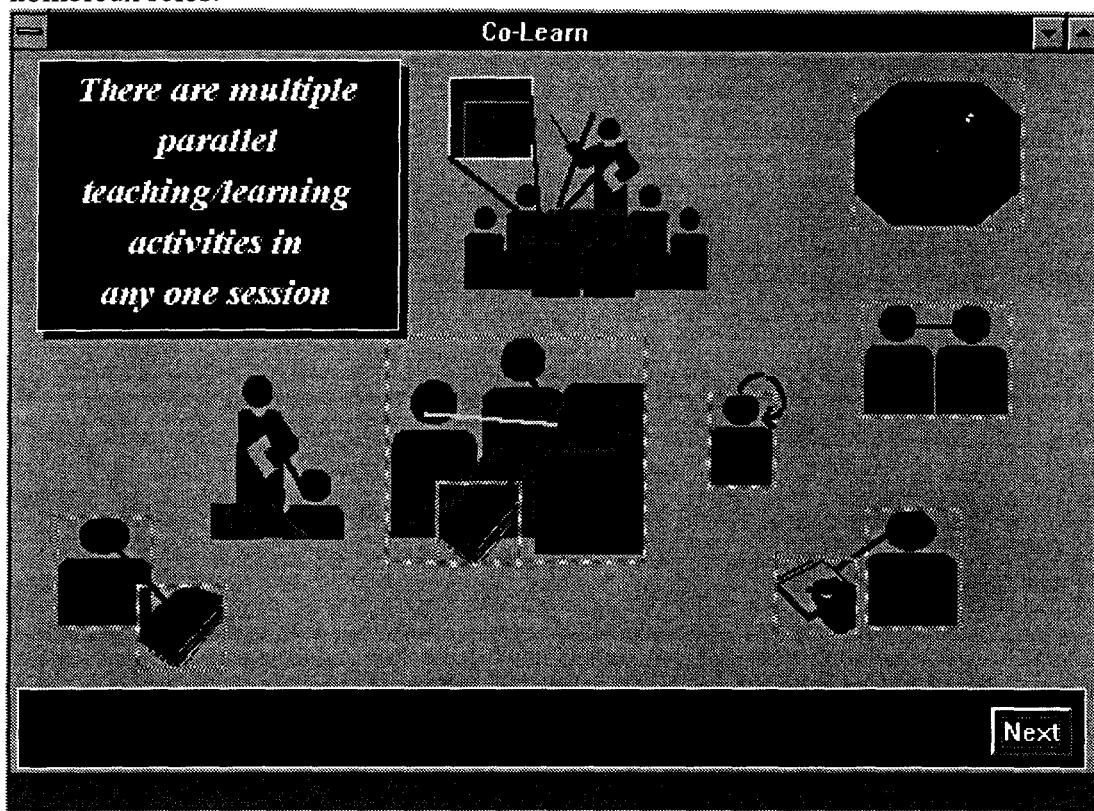


Figure 38. Les participants d'une réunion mènent de nombreuses activités [SMITH 94]

Le type de Salle Virtuelle d'Atelier de Travail que nous avons prédéfini comporte ainsi tous les outils généraux qui nous semblent utilisables dans une réunion [COLEARN 92c] [COLEARN 92f] [COLEARN 93f] : fenêtre publique, télépointeur, urne de vote, sablier, audio-réunion, outil de bavardage. Ceci permet à un formateur de disposer d'un exemple de salle virtuelle très riche, qui lui donne une idée des possibilités du système.

Le formateur peu rompu aux Nouvelles Technologies de l'Education, à l'enseignement à distance ou à l'apprentissage coopératif peut ôter certains outils pour obtenir un environnement technique plus simple à gérer. Il peut aussi choisir des rôles qui lui permettent de mieux contrôler le comportement du groupe.

Une autre solution consiste à utiliser un agent informatique jouant le rôle d'assistant d'animation. En raison du temps disponible dans le cadre du projet pour concevoir, réaliser et évaluer la plate-forme, nous n'avons pas pu réaliser cet agent. Nous ne le décrivons pas dans ce chapitre mais dans le chapitre concernant les extensions de Co-Learn. Le lecteur intéressé pourra aussi se reporter à [HOOGS 92b].

- La Salle Virtuelle de Forum permet à un groupe éventuellement large de coopérer en temps différé. Ce type de Salle Virtuelle supporte une coopération centrée sur la conversation.

On peut envisager plusieurs utilisations de ce type de Salle Virtuelle prédéfini.

- Les participants peuvent l'utiliser uniquement pour coordonner leurs actions qui ne sont pas supportées par l'informatique. Ils utilisent alors les outils de conversation, de gestion des tâches et de gestion des actions [COLEARN 92d][COLEARN 93e] pour organiser les tâches et négocier leurs engagements. Ils peuvent aussi utiliser l'outil [COLEARN 93b] d'accès à la bibliothèque de documents hypermédia [COLEARN 93c] pour y stocker les documents liés à leurs activités non informatisées : cahiers des charges, rapports d'activité... Ils peuvent inclure des références à ces documents dans leurs conversations.

- Les participants peuvent aussi utiliser ce type de Salle Virtuelle pour mener sur une longue période un débat scientifique, argumenté et illustré par des références à des documents scientifiques. Ils utilisent dans ce cas l'outil de conversation. Ils exploitent également l'outil d'accès à la médiathèque pour retrouver ou stocker les documents scientifiques qu'ils peuvent référencer. Au cours de leur coopération, les participants peuvent élaborer un rapport scientifique qu'ils partagent par l'intermédiaire de la médiathèque.

1.1.3. Configurer une Salle Virtuelle

Pour configurer une Salle Virtuelle, les formateurs peuvent intervenir sur plusieurs paramètres :

- la liste des participants autorisés à entrer,
- la liste des outils présents,
- les rôles fonctionnels affectés aux participants dans les activités supportées par les outils,
- les modes de médiation des activités supportées par les outils.

Chacun des quatre types prédéfinis dans Co-Learn est spécialisé pour un mode de communication : temps réel ou temps différé.

Cependant, il est possible de définir des types de salle virtuelle qui ne soient pas dédiés à un mode de communication donné mais permettent au contraire au groupe d'y tenir des réunions et à ses membres de venir y travailler individuellement tour à tour (exemple de la salle de garde ou de permanence tuteur).

La salle virtuelle est toutefois quand même caractérisée soit par la tâche que le groupe doit y mener et pour laquelle il l'a créée, soit par l'équipe qui en a l'usage exclusif, soit

par le mode de coopération qui peut être défini. Il est important que la Salle Virtuelle ait un caractère.

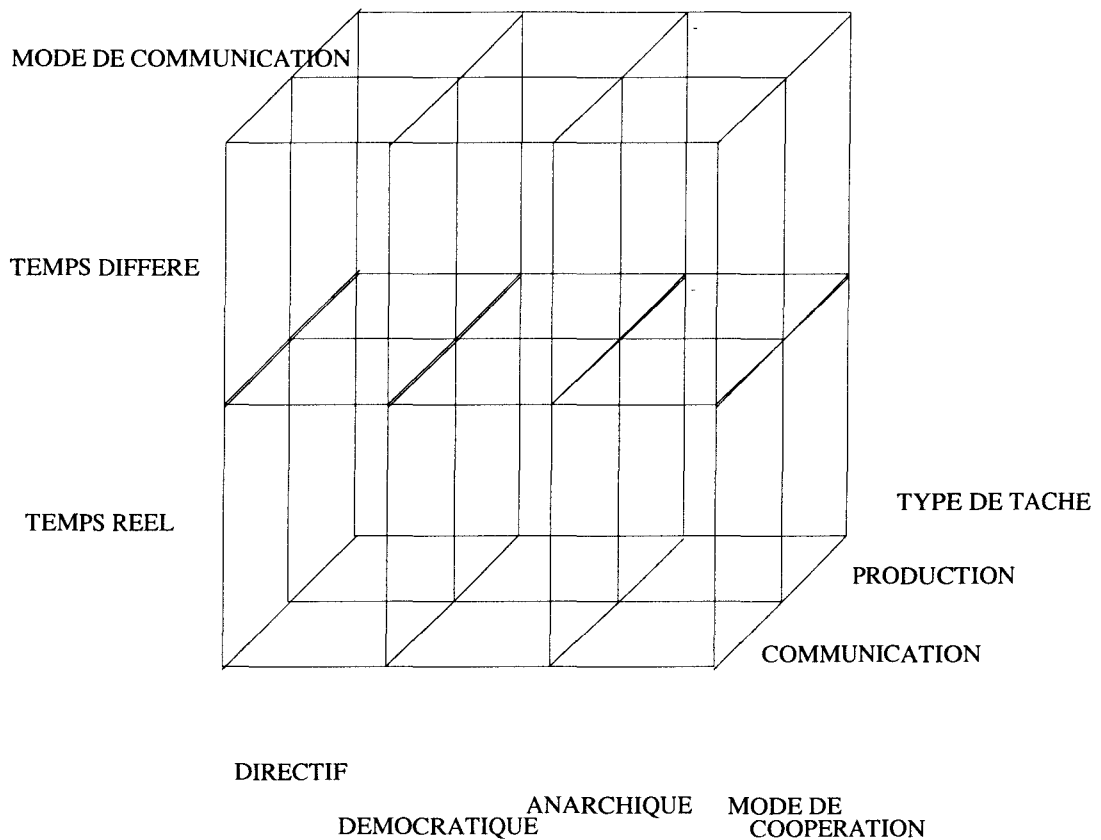


Figure 39. Caractérisation des espaces de coopération

Le mode de coopération peut être influencé par l'affectation des rôles fonctionnels mais aussi par les modes de médiation choisies pour les activités de la Salle Virtuelle. Le tableau qui suit donne quelques exemples de spécialisation de Salles Virtuelles.

Exemple	Caractérisation	Mode de coopération	Mode de Communication
Salle de Projet	Tâche	Multiple	Multiple
Amphithéâtre	Coopération Communication	Directif	Temps réel
Salle de Réunion	Communication	Multiple	Temps réel
Salle de Forum	Communication	Multiple	Temps différé
Salle de Travaux Pratiques	Coopération Communication	Assistance	Temps réel
Salle des professeurs	Equipe	Multiple	Multiple
Bibliothèque	Tâche	Opportuniste	Temps Différé
Salle de quart	Equipe Tâche	Multiple	Multiple
Salle de détente	Coopération	Convivial	Multiple
Salle de dessin	Tâche	Multiple	Multiple

Table 10. Exemple de spécialisations de Salles Virtuelles

1.2. Concevoir les activités de coopération

Nous avons développé pour Co-Learn une méthode de conception des activités de coopération à partir de la méthode AMIGO. Nous avons conçu deux variations pour les activités en temps réel d'une part et les activités en temps différé d'autre part. Dans les deux variations, nous concilions l'approche orientée rôle et l'approche orientée médiateur de l'analyse des activités de groupe identifiées par [PANKO 89c]. Cette conciliation permet de profiter des avantages des deux approches et d'atténuer les inconvénients de chacune.

Dans l'approche orientée médiateur, les échanges entre deux membres du groupe sont médiatisés et coordonnés par une entité centrale qui assure la régulation de l'activité. L'échange direct entre deux membres n'intervient pas dans l'évolution de l'activité. Cette première approche est supportée par le modèle AMIGO. L'avantage de cette approche est de mettre l'accent sur le groupe dans son ensemble et de donner une réelle existence à l'activité, à travers son médiateur.

In the foreground of this mediator oriented approach stands the group and the Activity. The Activity, through its Mediator, becomes an acting entity of its own, having a real distinct existence.

[Ibid., p. 273]

De plus, cette approche centralisée facilite la synchronisation et la coordination.

*The power of a mediator oriented approach to automating Activity execution lies in the fact that **synchronisation** and **co-ordination** conditions can be treated easily.*

[Ibid., p. 274]

L'inconvénient principal de cette approche est qu'elle n'assiste pas les participants dans leur rôle.

The major disadvantage of this approach lies in particular in the lack of individual support for human Actors playing Roles in the Activity.

[Ibid., p. 274]

L'approche orientée rôle considère une activité comme un ensemble d'acteurs distribués qui appliquent localement des règles de régulation pour la coopération. Le principal avantage de cette approche est la conception d'une assistance locale aux participants pour se comporter coopérativement. [PANKO 89c] conclue qu'il serait intéressant d'adopter AMIGO pour permettre une approche orientée rôle. Il remarque que l'architecture proposée pour AMIGO par [PAYS 89] permettrait de supporter l'implantation des deux approches. Nous pensons que notre proposition s'inscrit dans la continuité des travaux du groupe AMIGO.

1.2.1. Concevoir les activités de coopération en temps réel : exemple de l'Activité Partage de Ressource

Nous décrivons dans ce paragraphe l'adaptation que nous avons apportée au modèle AMIGO pour concevoir les activités de coopération en temps réel. Nous appuyons cette description sur l'exemple de la conception de l'activité de partage de ressource. La méthode obtenue a été utilisée pour concevoir les activités de partage des outils de groupe en temps réel : la fenêtre publique, le télépointeur, l'outil de vote, le chronomètre, l'audioconférence, le messenger.

L'activité de partage de ressource a pour objet d'assister la coordination d'un groupe qui partage une ressource unique (par exemple un télépointeur ou un tour de parole). Nous décrivons ici deux modes de médiation proposés de base par Co-Learn. Dans les deux modes, l'Agent de Médiation accorde la ressource à tout utilisateur qui la demande dès l'instant qu'elle est libre. Si la ressource est utilisée par un participant, seuls les participants autorisés à émettre une réquisition ou une attribution peuvent la prendre. Dans le mode *demande mémorisée*, l'Agent de Médiation permet de demander la ressource quand elle est utilisée pour l'obtenir dès que le détenteur la libérera. Il faut noter que ces deux modes de médiation permettent de réaliser de nombreux modes de coopération selon la distribution des rôles. Par exemple, en affectant le rôle de superviseur à tous les participants, on permet un mode de

coopération anarchique (chacun peut prendre la ressource à tout moment, quitte à la confisquer à son détenteur); en affectant le rôle de compétiteur à tous, on réalise un mode de coopération poli (on ne peut prendre la ressource que si personne ne l'utilise) et confiant (on suppose que personne ne monopolisera la ressource). Nous allons voir qu'il est également possible de modifier le mode de médiation au cours de l'activité. Pour des cas plus spécifiques, il est facile de définir de nouveaux modes de médiation et de nouveaux rôles.

Ci-après, nous développons d'abord une analyse AMIGO de l'activité de partage de ressource qui nous amène à proposer l'ajout d'un nouveau composant au modèle. Ensuite, nous développons l'analyse de cette même activité avec le modèle Co-Learn. Le but du développement de ces deux analyses est d'expliquer et d'illustrer comment le modèle Co-Learn adapte le modèle AMIGO pour permettre de concevoir des outils qui supportent les activités de groupe en temps réel.

a) Analyse de type AMIGO de l'activité de partage de ressource et identification d'un nouveau composant

Nous ne proposons ici que des extraits de la conception de l'activité de partage de ressource selon la méthode AMIGO. L'analyse complète se trouve dans l'annexe II.

(1) Le composant Rôles

Rôles : Superviseur, Compétiteur

SUPERVISEUR	
identité, adresse	gérées par la Salle Virtuelle
Types de messages envoyés	Demande, Réquisition, Attribution, Libération
Types de messages reçus	Demande, Réquisition, Attribution, Libération
Attributs	état

état \in { indifférent, possesseur, candidat }

COMPETITEUR	
identité, adresse	gérées par la Salle Virtuelle
Types de messages envoyés	Demande, Libération
Types de messages reçus	Demande, Libération, Réquisition, Attribution
Attributs	état

état \in { indifférent, possesseur, candidat }

AGENT DE PARTAGE DE RESSOURCE	
identité, adresse	gérées par la Salle Virtuelle
Types de messages envoyés	Demande, Réquisition, Attribution, Libération
Types de messages reçus	Demande, Réquisition, Attribution, Libération
Attributs	détenteur candidat

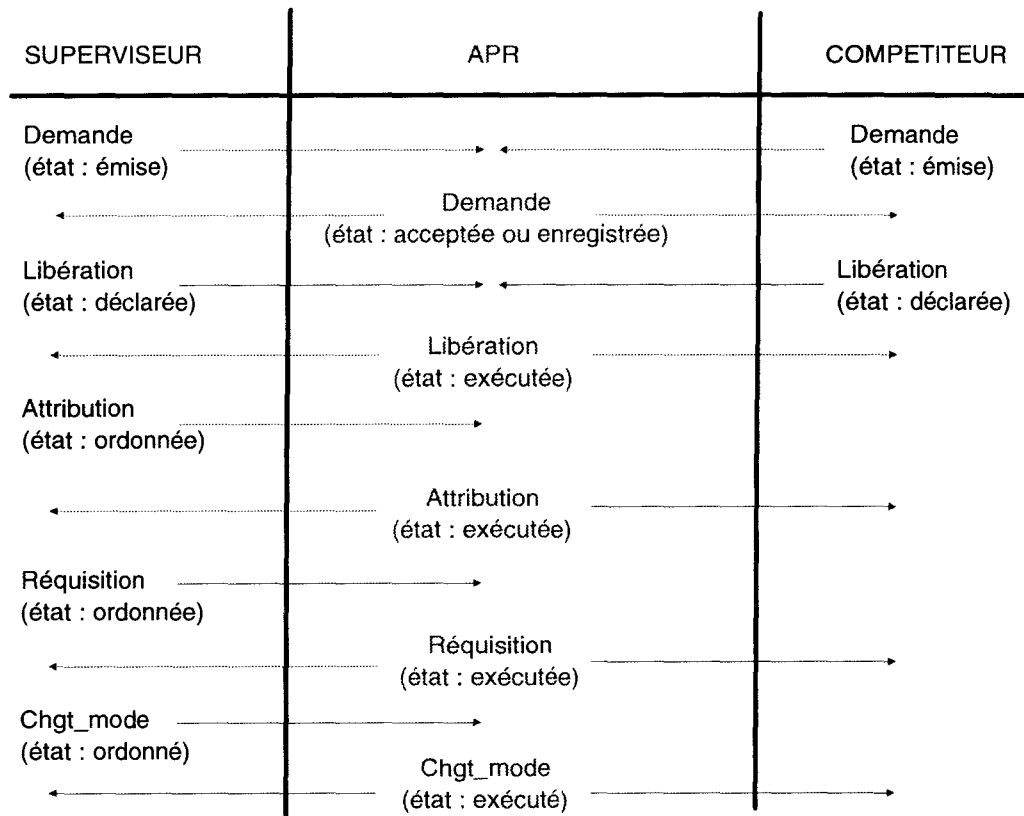


Figure 40. La communication entre les entités de l'Activité de Partage de Ressource selon AMIGO

(2) Le Composant Messages

DEMANDE			
Etat	EMISE	ACCEPTEE	ENREGISTREE
Requête	DECIDE	EXECUTE	
Expéditeur	SUPERVISEUR COMPETITEUR	APR	APR
Destinataire	APR	SUPERVISEUR COMPETITEUR	SUPERVISEUR COMPETITEUR
Candidat	(Expéditeur)	Identifiant du nouvel utilisateur de la ressource	Identifiant du participant dont la demande est temporisée

CHGT_MODE		
Etat	ORDONNE	EXECUTE
Requête	DECIDE	
Expéditeur	SUPERVISEUR	APR
Destinataire	APR	SUPERVISEUR COMPETITEUR
Mode	Nouveau mode de partage	Nouveau mode de partage

Nous remarquons une première différence par rapport au modèle AMIGO. Les messages utilisés dans les activités de groupe en temps réel ne sont pas persistants, comme dans le cas des activités de communication asynchrone : ils ne sont pas stockés dans des boîtes à lettres de groupe ou personnelles. Ils ne sont pas référencés par d'autres objets. C'est pourquoi ils n'ont pas d'identifiant.

(3) Le Composant Fonctions

Fonctions génériques :

DIFFUSE *un message*

Exécutée par l'APR, pour diffuser un message à tous les participants de l'activité

ENVOIE *un message A un participant*

Exécutée par l'APR pour envoyer un message à un participant particulier de l'activité

GEN_MSG DU TYPE *un type de message* DANS L'ETAT *un état*

Génère un message du type approprié dans l'état précisé. L'état doit être défini dans la définition de ce type de message.

TRANSFERE(msg1,msg2)

Copie la valeur des attributs du message msg1 vers les mêmes attributs du message msg2.

AFFECTE *rôle.attribut = valeur*

Assigne une valeur à l'attribut d'une instance de rôle.

PASSE *un participant* DANS L'ETAT *un état*

Exécutée par l'APR pour modifier l'état du rôle d'un participant dans l'activité

PASSE EN PHASE *une phase*

Exécutée par l'APR pour modifier la phase de l'activité

PASSE EN MODE *un mode*

Exécutée par l'APR pour modifier le mode de gestion de la coopération

(4) Le composant Règles

Mode amnésique

Règles suivies par l'Agent de Partage de Ressource

DEBUT DE PHASE libre

A RECEPTION D'UNE Demande émise FAIRE

PASSE Demande.expéditeur DANS L'ETAT possesseur

msg1 = GEN_MSG Demande DANS L'ETAT acceptée

msg1.candidat = Demande.expéditeur

DIFFUSE msg1

```

PASSE EN PHASE utilisée
A RECEPTION D'UNE Réquisition ordonnée FAIRE
  PASSE Réquisition.expéditeur DANS L'ETAT possesseur
  msg1 = GEN_MSG Réquisition DANS L'ETAT exécutée
  msg1.autorité = Réquisition.expéditeur
  DIFFUSE msg1
  PASSE EN PHASE utilisée
A RECEPTION D'UNE Attribution ordonnée FAIRE
  PASSE Attribution.élu DANS L'ETAT possesseur
  msg1 = GEN_MSG Attribution DANS L'ETAT exécutée
  TRANSFERE(Réquisition,msg1)
  DIFFUSE msg1
  PASSE EN PHASE utilisée
A RECEPTION D'UN Chgt_mode ordonné FAIRE
  PASSE EN MODE Chgt_mode.mode
  msg1 = GEN_MSG Chgt_mode DANS L'ETAT exécuté
  TRANSFERE(Chgt_mode,msg1)
  DIFFUSE msg1
FIN DE PHASE libre

DEBUT DE PHASE utilisée
A RECEPTION D'UNE Réquisition ordonnée FAIRE
  PASSE Réquisition.expéditeur DANS L'ETAT possesseur
  msg1 = GEN_MSG Réquisition DANS L'ETAT exécutée
  msg1.autorité = Réquisition.expéditeur
  DIFFUSE msg1
A RECEPTION D'UNE Attribution ordonnée FAIRE
  PASSE détenteur DANS L'ETAT indifférent
  PASSE Attribution.élu DANS L'ETAT possesseur
  msg1 = GEN_MSG Attribution DANS L'ETAT exécutée
  TRANSFERE(Réquisition,msg1)
  DIFFUSE msg1
A RECEPTION D'UNE Libération déclarée FAIRE
  PASSE Libération.expéditeur DANS L'ETAT indifférent
  msg1 = GEN_MSG Libération DANS L'ETAT transmise
  TRANSFERE(libératoion,msg1)
  DIFFUSE msg1
  PASSE EN PHASE libre
A RECEPTION D'UN Chgt_mode ordonné FAIRE
  PASSE EN MODE Chgt_mode.mode
  msg1 = GEN_MSG Chgt_mode DANS L'ETAT exécuté
  TRANSFERE(Chgt_mode,msg1)
  DIFFUSE msg1
FIN DE PHASE utilisée

```

Nous ajoutons au modèle d'analyse de l'activité AMIGO un Composant Droits. Ce composant est lié à la notion de phase de l'activité, qui existe dans le modèle AMIGO et à celle d'état du rôle qui est spécifique au modèle Co-Learn (l'état du rôle peut être vu comme un attribut particulier défini par Co-Learn pour tous les types de rôle des activités de groupe en temps réel).

Le composant Droits que nous proposons précise l'évolution des droits de chaque rôle en fonction de la phase de l'activité et de l'état du rôle. Ce composant pourrait être intégré à AMIGO soit dans le composant Règle, soit dans le composant Rôle, mais il n'est pas évoqué dans [DANIE 89]. Ce composant droits introduit des fonctions génériques qui n'existent pas dans le modèle AMIGO :

RAZ DES DROITS

Exécutée par un agent de rôle pour supprimer tous les droits de l'utilisateur

AUTORISE *liste de messages*

Exécutée par un agent de rôle pour définir la liste des messages que l'utilisateur peut envoyer. Certains types de messages permettent à l'utilisateur d'agir, donc de modifier l'état ou les données de l'activité : ils sont rendus accessibles à l'utilisateur par les Ressources d'Action. Les autres types de messages sont produits par les Ressources d'Information et permettent à l'utilisateur de visualiser les données de l'activité.

(5) Règles de Conversation suivies par Les Rôles

Mode amnésique

Règles suivies par un Superviseur

```

DEBUT DE PHASE libre
RAZ DES DROITS
AUTORISE Demande,Attribution,Réquisition,Chgt_mode
A RECEPTION D'UN Chgt_état FAIRE
    Chgt_état = état
A RECEPTION D'UN Chgt_phase FAIRE
    PASSE EN PHASE Chgt_phase.phase
FIN DE PHASE libre
  
```

```

DEBUT DE PHASE utilisée
RAZ DES DROITS
SELON état FAIRE
    indifférent : AUTORISE Réquisition,Attribution,Chgt_mode
    possesseur : AUTORISE Libération,Attribution,Chgt_mode
A RECEPTION D'UN Chgt_état FAIRE
    état = Chgt_état.état
RAZ DES DROITS
SELON état FAIRE
    indifférent : AUTORISE Réquisition, Attribution,Chgt_mode
    possesseur : AUTORISE Libération, Attribution,Chgt_mode
A RECEPTION D'UN Chgt_phase FAIRE
    PASSE EN PHASE Chgt_phase.phase
FIN DE PHASE utilisée

```

Le développement du composant droits fait apparaître un bloc d'instructions du type :

```

RAZ DES DROITS
SELON état FAIRE
    état1 : AUTORISE liste 1 de messages
    ....
    étatN : AUTORISE liste N de messages

```

Il est logique de remplacer ce bloc par une fonction générique EVALUE LES NOUVEAUX DROITS. C'est ce que Co-Learn propose. En introduisant cette fonction, on remarque que les règles suivies par les rôles ne changent entre les rôles et entre les modes que dans la réalisation de la fonction EVALUE LES NOUVEAUX DROITS.

Nous montrons dans l'analyse suivante qu'il est intéressant d'utiliser le même type d'agent de rôle pour tous les rôles et tous les modes et de le configurer par le chargement d'un composant définissant le comportement de la fonction EVALUE LES NOUVEAUX DROITS.

b) Utilisation du modèle Co-Learn pour la conception de l'activité de partage de ressource

Pour garantir la malléabilité de la gestion de la coopération, les activités de coopération de groupe en temps réel de l'environnement Co-Learn utilisent plusieurs messages génériques :

Chgt_règles

Chgt_phase

Chgt_état

Chgt_données

L'utilisation de ces messages permet de changer le mode de coopération dynamiquement.

En effet, les Agents de Rôle ne sont pas spécialisés pour un mode de coopération : ils reçoivent les Règles de Conversation qu'ils doivent suivre de la part de l'Agent de Médiation de l'Activité, dans un message Chgt_règles. Ces règles précisent les droits d'accès du participant aux Ressources d'Action et aux Ressources d'Information en fonction de la phase de l'activité et de l'état du rôle du participant. Les Ressources d'Action sont les messages que les participants peuvent envoyer. Les Ressources d'Information sont les données qu'ils peuvent consulter. A la réception des messages Chgt_phase et Chgt_état, les Agents de Rôle utilisent la fonction générique *EVALUE LES NOUVEAUX DROITS* qui consulte les règles pour déterminer les nouveaux droits du participant dans l'activité.

Le Message Chgt_donnée permet d'initialiser ou de modifier la valeur d'une donnée.

Pour gérer les entrées et sorties des participants, les activités de groupe en temps réel utilisent également les messages génériques Arrivée et Départ. Le comportement générique d'une activité de groupe en temps réel à la réception d'un message Arrivée consiste à envoyer au nouvel arrivant :

- un message Chgt_règles pour lui communiquer les règles de conversation adaptées à son rôle et au mode de coopération courant de l'activité,
- un message Chgt_état et un message Chgt_phase pour lui communiquer l'état de son rôle, tel qu'il était lorsque le participant a quitté l'activité à la session précédente et la phase que l'activité a atteinte durant l'absence du participant.

L'Agent de Médiation envoie un message Chgt_règles approprié à chaque participant à la réception d'un message Chgt_mode.

Cette conception des activités de groupe en temps réel permet de limiter la charge de raisonnement de l'Agent de Médiation. Ainsi, il envoie à tous les agents de rôle les mêmes messages de changement de phase et de changement de données tandis que chaque Agent de Rôle l'exploite en fonction des règles que l'Agent de Médiation lui a communiquées.

Une sous-classe générique d'activité de groupe définit le comportement générique des activités qui nécessitent de communiquer la liste des participants de l'activité aux autres participants. En plus de respecter le comportement générique d'une activité de groupe, ces activités avertissent les autres participants de l'arrivée et du départ des autres participants.

Ci-après, nous présentons des partiellement l'analyse de l'activité de partage de ressource selon la méthode Co-Learn que nous avons développée. La totalité de l'analyse est fournie en annexe III.

(1) Le composant Rôles

Agents de Rôles : Superviseur, Compétiteur

SUPERVISEUR	
identité, adresse	géré par la Salle Virtuelle
Types de messages envoyés	Demande, Réquisition, Attribution, Libération
Types de messages reçus	Chgt_règles, Chgt_phase, Chgt_état, Chgt_donnée
Attributs	état phase règles données

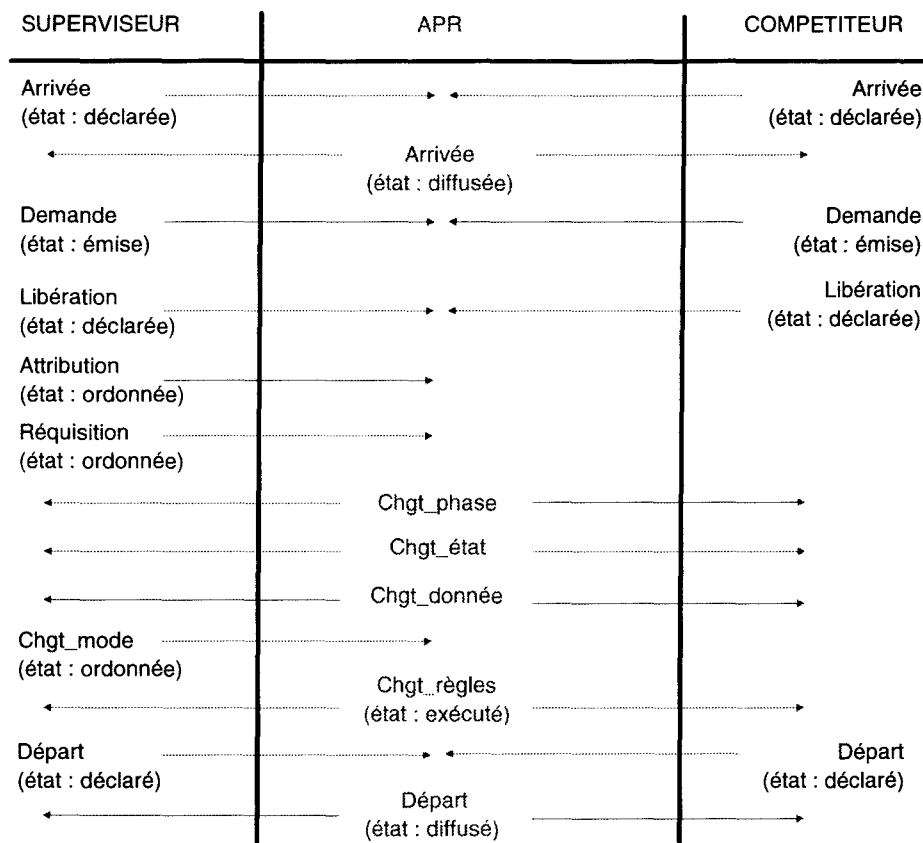
état ∈ { indifférent, possesseur, candidat }

phase ∈ { libre, utilisée, convoitée }

COMPETITEUR	
identité, adresse	géré par la Salle Virtuelle
Types de messages envoyés	Demande, Libération
Types de messages reçus	Chgt_règles, Chgt_phase, Chgt_état, Chgt_donnée
Attributs	état phase règles données

état ∈ { indifférent, possesseur, candidat }

phase ∈ { libre, utilisée, convoitée }



Agent de Médiation : Agent de Partage de Ressource

AGENT DE PARTAGE DE RESSOURCE	
identité, adresse	gérées par la Salle Virtuelle
Types de messages envoyés	Chgt_règles, Chgt_phase, Chgt_état, Chgt_donnée
Types de messages reçus	Demande, Réquisition, Attribution, Libération, Chgt_mode
Attributs	données phase mode détenteur candidat

(2) Le Composant Messages

Types de messages génériques des activités de groupe en temps réel définis par Co-Learn

ARRIVEE		
Etat	DECLAREE	TRANSMISE
Requête	TRANSMETTRE	
Expéditeur	Agent de Rôle	Agent de Médiation
Destinataire	Agent de Médiation	Agent de Rôle

DEPART		
Etat	DECLARE	TRANSMIS
Requête	TRANSMETTRE	
Expéditeur	Agent de Rôle	Agent de Médiation
Destinataire	Agent de Médiation	Agent de Rôle
Sortant	(Expéditeur)	Identifiant du participant qui quitte l'activité

CHGT_MODE	
Etat	ORDONNE
Requête	EXECUTER
Expéditeur	Agent de Rôle autorisé
Destinataire	Agent de Médiation
Mode	nouveau mode de gestion de la coopération

CHGT_PHASE	
Etat	ORDONNE
Requête	EXECUTER
Expéditeur	Agent de Médiation
Destinataire	Agent de Rôle
Mode	Nouvelle phase de l'activité

CHGT_ETAT	
Etat	ORDONNE
Requête	EXECUTER
Expéditeur	Agent de Médiation
Destinataire	Agent de Rôle
Etat	Nouvel état du rôle

CHGT_DONNEE	
Etat	NOTIFIE
Requête	MEMORISER
Expéditeur	Agent de Médiation
Destinataire	Agent de Rôle
Identifiant	Identifiant de la donnée à modifier
Valeur	Nouvelle valeur de la donnée

CHGT_REGLES	
Etat	ORDONNE
Requête	EXECUTER
Expéditeur	Agent de Médiation
Destinataire	Agent de Rôle
Règles	Nouvelles règles de conversation du rôle

Types de message spécifiques aux Activités de Partage de Ressource

DEMANDE	
Etat	EMISE
Requête	DECIDE
Expéditeur	SUPERVISEUR COMPETITEUR
Destinataire	APR

Les autres types de message sont spécifiés dans l'annexe III.

(3) Le Composant Fonctions**Fonctions génériques des activités de groupe en temps réel proposées par Co-Learn**

EVALUE des règles POUR un type de rôle DANS un mode

Exécutée par l'APR pour déterminer les règles de conversation que doit respecter le type de rôle concerné dans le mode précisé.

EVALUE NOUVEAUX DROITS

Exécutée par un rôle pour mettre à jour les droits de l'utilisateur en exécutant les règles de conversation.

PASSE EN MODE un mode

Exécutée par l'APR pour modifier le mode de gestion de la coopération et déterminer les nouvelles règles de médiation qu'il doit respecter. L'APR provoque également par cette fonction l'envoi à chaque participant de l'activité les nouvelles règles de conversation adaptées à son rôle.

```
POUR CHAQUE membre DANS {superviseur, compétiteur} FAIRE
  EVALUE règles POUR membre DANS un mode
  GEN_MSG Chgt_règles
  Chgt_règles.règles = règles
  DIFFUSE Chgt_règles A membre
```

PASSE DANS L'ETAT un état

Exécutée par un agent de rôle pour évaluer, à l'aide des règles de conversation qui lui ont été assignées, les nouveaux droits de l'utilisateur dans l'activité.

Procédure :

```
état = un état
EVALUE LES NOUVEAUX DROITS
```

PASSE EN PHASE une phase

Exécutée par un agent de rôle pour évaluer, à l'aide des règles de conversation qui lui ont été assignées, les nouveaux droits de l'utilisateur dans l'activité.

Procédure :

phase = une phase
EVALUE LES NOUVEAUX DROITS

PASSE un participant DANS L'ETAT un état

Exécutée par l'APR pour mémoriser le nouvel état du rôle du participant dans l'activité et communiquer ce changement au participant.

Procédure :

un participant.état = un état
GEN_MSG Chgt_état
Chgt_état.Etat = un état
ENVOIE Chgt_Etat A un participant

AFFECTE une valeur A un identifiant

Exécutée par un Agent de Rôle pour mettre à jour la valeur d'une donnée de l'activité. L'Agent de Rôle évalue les règles pour déterminer si l'état du rôle et la phase de l'activité autorise la communication de la donnée à l'utilisateur (par l'intermédiaire d'une Ressource d'Information).

Fonctions génériques identifiées par AMIGO et modifiées par Co-Learn pour les activités de groupe en temps réel

PASSE EN PHASE une phase

Exécutée par l'APR pour modifier et mémoriser la phase de l'activité et diffuser ce changement de phase à tous les participants de l'activité présents dans la salle virtuelle.

Cette fonction appelle les fonctions : GEN_MSG et DIFFUSE A et utilise le type de message Chgt_phase.

La spécification des Fonctions génériques identifiées par AMIGO que Co-Learn utilise sans les modifier peut être consultée dans l'annexe III.

Fonctions spécifiques aux Activités de Partage de Ressource

DONNE A un participant

Exécutée par l'APR pour modifier, mémoriser et diffuser l'identité du nouveau détenteur de la ressource.

Procédure :

détenteur = Réquisition.expéditeur

```

PASSE détenteur DANS L'ETAT possesseur
GEN_MSG Chgt_donnée
Chgt_donnée.identifiant = 1
Chgt_donnée.valeur = détenteur
DIFFUSE Chgt_donnée

```

(4) Le composant Règles

Nous ne présentons ici du composant règle que la partie liée au mode amnésique. Cependant, il est présenté dans sa totalité dans l'annexe III.

Règles suivies par l'Agent de Partage de Ressource

MODE amnésique (l'APR ne mémorise pas de candidature)

DEBUT DE PHASE libre

```

A RECEPTION D'UNE Arrivée déclarée FAIRE
    EVALUE règles POUR Arrivée.expéditeur DANS mode
    GEN_MSG Chgt_règles
    Chgt_règles.règles = règles
    ENVOIE Chgt_règles A Arrivée.expéditeur
    GEN_MSG Chgt_état
    Chgt_état.état = Arrivée.expéditeur.état
    ENVOIE Chgt_État A Arrivée.expéditeur
    GEN_MSG Chgt_phase
    Chgt_phase.phase = "libre"
    ENVOIE Chgt_Phase A Arrivée.expéditeur
A RECEPTION D'UNE Demande FAIRE
    DONNE A Demande.expéditeur
    PASSE EN PHASE utilisée
A RECEPTION D'UNE Réquisition FAIRE
    DONNE A Réquisition.expéditeur
    PASSE EN PHASE utilisée
A RECEPTION D'UNE Attribution FAIRE
    DONNE A Attribution.élu
    PASSE EN PHASE utilisée
A RECEPTION D'UN Chgt_mode FAIRE
    PASSE EN MODE Chgt_mode.mode

```

FIN DE PHASE libre

DEBUT DE PHASE utilisée

```

A RECEPTION D'UNE Arrivée déclarée FAIRE
    EVALUE règles POUR Arrivée.expéditeur DANS mode
    GEN_MSG Chgt_règles
    Chgt_règles.règles = règles
    ENVOIE Chgt_règles A Arrivée.expéditeur
    GEN_MSG Chgt_état
    Chgt_état.état = Arrivée.expéditeur.état
    ENVOIE Chgt_État A Arrivée.expéditeur
    GEN_MSG Chgt_phase

```



```

    Chgt_phase.phase = "utilisée"
    ENVOIE Chgt_Phase A Arrivée.expéditeur
A RECEPTION D'UNE Réquisition FAIRE
    PASSE détenteur DANS L'ETAT "indifférent"
    DONNE A Réquisition.expéditeur
A RECEPTION D'UNE Attribution ordonnée FAIRE
    PASSE détenteur DANS L'ETAT "indifférent"
    DONNE A Attribution.élu
A RECEPTION D'UNE Libération déclarée FAIRE
    PASSE détenteur DANS L'ETAT "indifférent"
    PASSE EN PHASE libre
A RECEPTION D'UN Chgt_mode FAIRE
    PASSE EN MODE Chgt_mode.mode
A RECEPTION D'UN Départ déclaré FAIRE
    msg1 = GEN_MSG DU TYPE Départ DANS L'ETAT transmis
    msg1.sortant = Départ.expéditeur
    DIFFUSE msg1
    SI (Départ.expéditeur == détenteur) ALORS
        PASSE détenteur DANS L'ETAT "indifférent"
        PASSE EN PHASE libre
FIN DE PHASE utilisée

```

Règles génériques définies par Co-Learn suivies par les Agents de Rôle des Activités en temps réel:

```

A RECEPTION D'UN Chgt_Règles FAIRE
    CHARGE REGLES Chgt_Règles.règles
A RECEPTION D'UN Chgt_état FAIRE
    PASSE DANS L'ETAT Chgt_état.état
A RECEPTION D'UN Chgt_phase FAIRE
    PASSE EN PHASE Chgt_phase.phase
A RECEPTION D'UN Chgt_donnée FAIRE
    AFFECTE Chgt_donnee.valeur A Chgt_donnee.identifiant

```

(5) Le Composant Droits

Règles envoyées par l'APR aux rôles :

Pour le mode amnésique

A destination des Superviseurs :

```

SELON (PHASE, ETAT)
    (libre,indifférent) : autorise(Demande, Attribution, Réquisition,Chgt_mode )
    (utilisée, indifférent) : autorise(Réquisition, Attribution,Chgt_mode)
    (utilisée,possesseur) : autorise(Libération, Attribution,Chgt_mode)

```

A destination des Compétiteurs

SELON (PHASE, ETAT)
 (libre,indifférent) : autorise(Demande)
 (utilisée,possesseur) : autorise(Libération)

c) Conclusion

Le modèle Co-Learn enrichit le modèle AMIGO pour permettre d'analyser et concevoir des activités de groupe en temps réel, qu'elles soient **instrumentales, de communication ou de coordination**. Ceci est réalisé par l'introduction du composant **Droits**. Ce composant définit les **Règles de Conversation** qui régissent l'accès aux **Ressources d'Information** et aux **Ressources d'Action** proposées par les **Outils**. Ces règles de conversation sont mises en oeuvre par les **Agents de Rôle** délégués auprès des **Agents d'Interface**. Le modèle Co-Learn est rendu plus souple que le modèle AMIGO par la notion de **Mode**, qui peut être changé dynamiquement pour activer de nouvelles **Règles de Médiation** et de nouvelles Règles de Conversation, ce qui offre une **Malléabilité Fonctionnelle**. Pour assurer la gestion des droits en temps réel sur les réseaux, le **Protocole de Communication** entre les Agents de Rôle distribués sur les clients et les Agents de Médiation partagés sur les serveurs est **allégé** par l'utilisation de Messages de **Chargement de Règles**, de **Changement d'Etat** et de **Changement de Phase**. Dans les activités que nous avons développées l'évaluation des changements d'état est suffisamment simple pour être prise en charge par l'**Agent de Médiation**. L'utilisation de composants, de messages et de fonctions génériques génère une meilleure **Modularité**, induisant une **Malléabilité Structurale**.

1.2.2. Concevoir les activités de coopération en temps différé

Dans le domaine de la coopération en temps différé, nous adaptons le modèle AMIGO pour supporter la conception d'activités de conversation dans le cadre d'une conférence. Le but est de supporter des conversations de groupe. En effet, les conceptions existantes choisissent entre deux types d'activités de communication : les conférences et les conversations.

Le premier type d'activité de communication considéré est la conférence de groupe, dont la conception mène à un système de type "Bulletin board", tel que CRMM (Conférence Répartie en Mode Messagerie) [BRUN 87] ou COSY [PALME 92]. Dans cette situation, les participants **partagent une base de messages**

(ou de fiches) structurée enrichie éventuellement de liens hypertextuels. Le rôle du système est essentiellement de faciliter la gestion du réseau de messages et assister les membres du groupe dans la suivi de l'évolution du réseau. La destination du message peut donc être le groupe et tous les membres partagent une vue de l'activité du groupe. Les fonctions que doit offrir un tel système de conférence ont été répertoriées par [PALME 92] et pourront être ajoutées facilement à notre système.

Le deuxième type d'activité est la conversation. C'est en pensant à ce type d'activité que le modèle AMIGO a été conçu. Quel que soit leur type (orienté procédure, orienté formulaire, orienté structure de communication, orienté conversation), tous les modèles de communication de groupe répertoriés par [PRINZ 89] découpent l'activité de communication du groupe en communications interpersonnelles. Même si la conception du système nécessite que les concepteurs aient une vue globale de l'activité du groupe, les systèmes conçus ne restituent pas cette vue globale aux utilisateurs. Chaque membre n'a qu'une vue égocentrique, personnelle de sa participation à certaines activités du groupe. Il ne connaît pas les activités auxquelles participent les autres membres, il ne peut pas situer les activités auxquelles il participe par rapport à l'activité globale du groupe. Souvent un participant d'une conversation ne connaît même pas le rôle et les actions des autres participants de cette conversation. Le système donne à chacun les moyens de jouer son rôle, souvent sans en connaître les tenants et aboutissants.

En combinant les deux types d'activités (conférence et conversation), nous restituons la situation naturelle de communication de groupe.
--

Même si un message est souvent émis explicitement d'un membre à destination d'un autre membre (ou d'un sous-groupe), il est aussi souvent mis à la connaissance du groupe entier et les interlocuteurs parlent en sachant et en comptant que le groupe est témoin de ce qui est dit (ou en sera informé). Il est évidemment important que le(s) destinataire(s) explicitement désigné(s) soi(en)t spécialement averti(s) du message mais il est souvent tout aussi important que les autres membres du groupe puissent prendre connaissance de la conversation. Par exemple, les questions qui sont posées dans une réunion sont parfois adressées par un des membres à une personne en particulier. Mais en posant sa question dans le cadre de la réunion, il est clair que l'intervenant tient à ce que les autres membres du groupe puissent prendre connaissance de sa question. De même, la personne qui répond à la question peut s'adresser explicitement à celle qui l'a posée mais il sait qu'il doit répondre comme si chaque membre du groupe avait posé cette même question.

Ainsi, dans le système d'aide à la communication de groupe en temps différé que nous avons conçu, les participants mènent des conversations dans le cadre d'une conférence. Ils soumettent donc les messages au système de conférence. En général, les messages sont visibles par tous les membres du groupe. Cependant, la conférence est organisée en tâches et en conversations, qui permettent de limiter si nécessaire la portée d'un message. Ainsi, un participant soumet son message au groupe dans le cadre d'une tâche et d'une activité de conversation. La tâche décrit le but que poursuivent les activités de conversation qui y sont rattachées. La tâche définit aussi un délai pour atteindre ce but et un responsable. Le responsable de la tâche peut modifier les caractéristiques de la tâche. Il peut suspendre la tâche pendant une période durant laquelle les participants ne peuvent que consulter les conversations de la tâche, sans en modifier les objets (conversation, liens, messages). Par défaut, tous les participants de la conférence (ce sont tous les participants de la Salle Virtuelle) peuvent accéder à toutes les tâches. Dans le cadre d'une tâche, les participants peuvent mener des conversations de différents types : conversations libres, conversations pour l'action, conversations de vote, conversations pour l'organisation de réunions.

Au premier niveau, l'Agent de Gestion de la Conférence supporte l'activité des membres d'un groupe liée à la participation à la conférence. Il reçoit toutes les requêtes des participants. Lorsqu'il reçoit une requête dont le traitement dépend de la médiation d'une conversation, l'AGC coopère avec les Agents de Médiation des Activités de Conversation, qui sont au deuxième niveau du système.

L'activité de groupe supportée au premier niveau ressemble à celle que [PAYS 89] prend comme exemple d'application du modèle AMIGO. Les activités supportées au second niveau sont du type de l'activité que [WOITA 89] analyse à l'aide du même modèle. La table 11 clarifie la différence entre conférence et conversation en donnant quelques exemples de types de messages de conférence et de conversation.

Messages de Conversation	Conversation Libre : • Sujet • Commentaire	Conversation Pour l'Action : • Demande • Proposition • Accord • Refus • Abandon • Report • Compte-rendu	Conversation Pour la Décision : • Motion • Opinion • Conclusion • Abandon	Conversation Pour la Planification de Réunions : • Proposition de dates • Choix de dates • Convocation • Accord • Refus • Report • Annulation • Compte-Rendu
Messages de Conférence	<ul style="list-style-type: none"> • Créer, Modifier, Suspender, Débuter, Terminer, Supprimer une Tâche • Consulter les Messages Non-Lus ou Non-Validés d'une Tâche • Créer, Consulter une Conversation • Répondre à, Modifier, Supprimer, Valider, Lire un Message Multimédia de Conversation • Créer, Supprimer un Lien Associatif • Créer, Supprimer une Référence Hypermédia • Créer, Supprimer une Référence Textuelle 			

Table 11. Messages de Conférence et Messages de Conversation

a) Conception de l'activité de Conférence Asynchrone

Dans cette activité, similaire à une activité de "*Bulletin Board*" simplifiée, les membres du groupe manipulent un espace de conversation commun, composé de conversations, de messages et de liens². Un auteur conserve le droit de modifier les messages qu'il a créés jusqu'à ce qu'il décide de les valider pour qu'ils soient publiés. Les participants envoient des requêtes de consultation et de manipulation de conversations et de messages à l'Agent de Gestion de la Conférence.

L'AGC réagit à la réception d'une requête de consultation en envoyant à l'expéditeur des informations sur des messages ou des conversations de la conférence. Une requête de consultation peut demander la liste des messages d'une tâche que le participant n'a pas encore lus, la liste des messages d'une tâche que le participant a créés et n'a pas encore publiés, l'arbre des messages qui composent une conversation, les détails d'un message de la conférence, la liste des conversations dans une tâche. Pour obtenir les informations sur une Conversation qu'il transmet au participant, l'AGC communique avec l'Activité de Conversation concernée, qui lui donne les informations accessibles en fonction du Rôle du participant dans la Conversation.

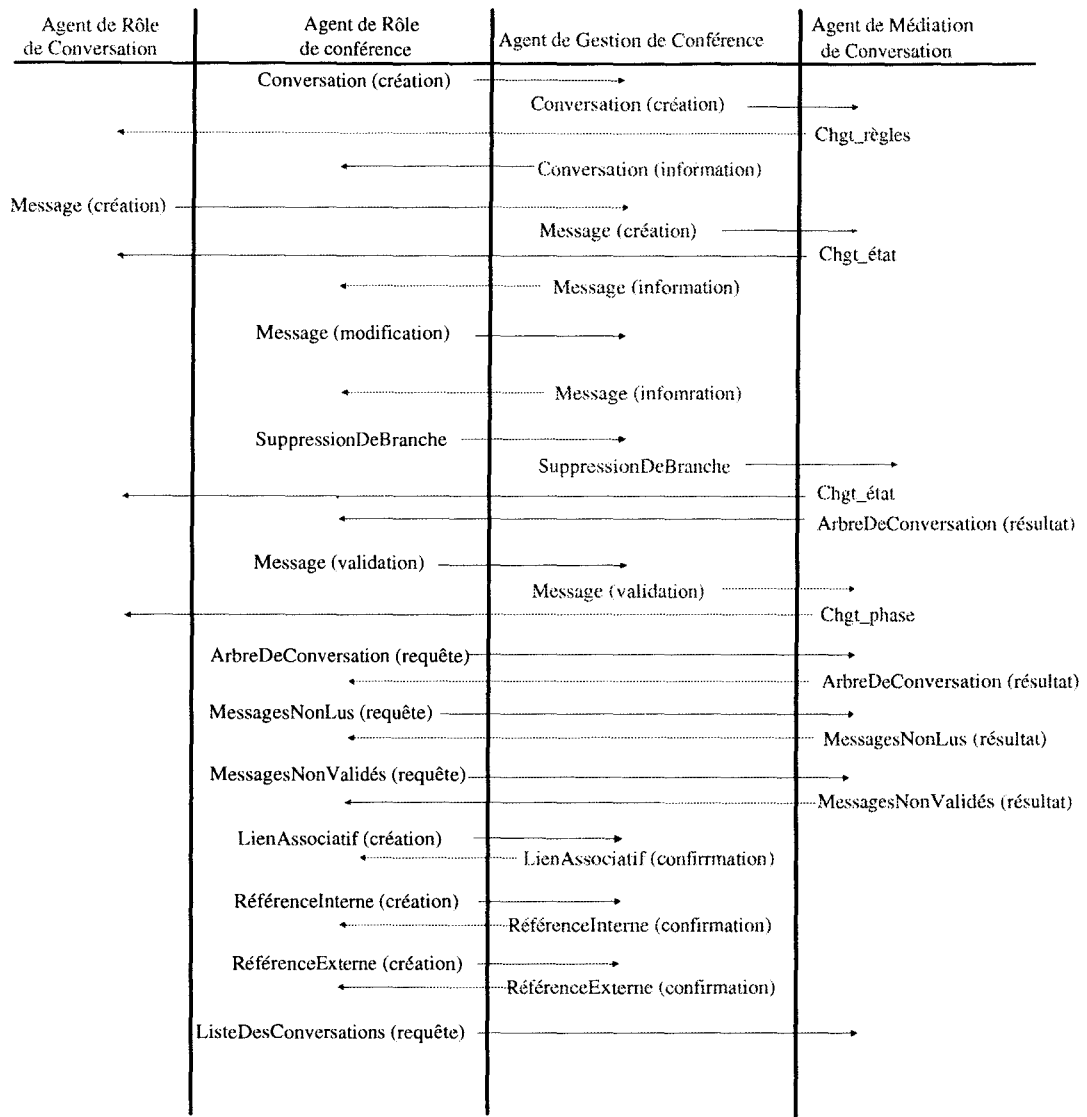
² L'activité de conférence que Claude Viéville et moi-même avons conçue pour Co-Learn permet également à différents rôles de gérer les tâches qui structurent la conférence. Cet aspect n'est pas développé ici, dans un souci de simplicité.

L'AGC réagit à la réception d'une requête en diffusant des informations sur la conférence aux participants concernés lorsque la requête reçue est une demande de modification des données de la conférence. Il faut noter qu'un participant ne peut pas modifier un message une fois qu'il l'a publié. Le seul type de requête adressée à l'AGC qui peut modifier les données de la conférence connues des autres participants est donc une demande de publication d'un message de la conférence. Même une demande de création de lien associatif ne modifie pas immédiatement les données de la conférence pour le reste du groupe. En effet, il n'est possible de créer un lien associatif qu'à partir d'un message qui n'est pas publié. Bien que la cible du lien puisse être un message déjà publié, le lien associatif n'est pas publié à sa création mais lorsque la source et la cible ont été publiées.

L'AGC communique avec une Activité de Conversation lorsqu'il reçoit une demande de modification des données sensibles de cette Conversation. Les données sensibles d'une Conversation sont les données dont la modification peut faire évoluer l'état de la Conversation ou l'Etat du Rôle d'un Participant dans la Conversation. Les types de requêtes dont le traitement nécessite une coopération de l'AGC avec une Activité de Conversation sont les demandes de création et les demandes de validation d'un message dans cette conversation. Les demandes de création font évoluer le Rôle de l'expéditeur dans la Conversation, les demandes de validation font évoluer à la fois le Rôle de l'expéditeur et l'Activité de Conversation.

Par exemple, lorsqu'un participant d'une conversation de vote demande à l'AGC de créer un message de type opinion, en réponse au message motion, son état passe à "votant". Dans cet état, le participant ne peut plus créer de nouveau message de type opinion mais l'Activité de Conversation n'évolue pas : rien ne change pour les autres participants. Lorsqu'un participant demande à l'AGC de valider le message de type opinion qu'il a créé, l'état du rôle du participant passe à "aVoté" et l'Activité de Conversation évolue puisque le résultat du vote est modifié.

L'AGC répond à une demande de conseil de réponse à un Message d'une Conversation par la liste des types de messages de conversation que l'expéditeur de la demande peut utiliser pour faire évoluer l'Activité de Conversation concernée. L'AGC obtient cette liste de types de messages auprès de l'Activité de Conversation concernée.



Dans certains échanges, l'AGC est le destinataire final des requêtes du participant : par exemple, lorsque le participant demande la liste des messages qu'il n'a pas publiés ou quand il demande à créer un lien associatif. Dans d'autres échanges, l'AGC est un intermédiaire entre le participant et une activité de conversation.

(1) Le composant Rôles

Les types de rôles intervenant dans une activité de conférence sont :

AGENT DE GESTION DE CONFERENCE (AGC)	
identité, adresse	gérées par la Salle Virtuelle
Types de messages envoyés	MessagesNonLus, MessagesNonValidés, ListeDesConversations, Conversation, ArbreDeConversation, Message, LienAssociatif, RéférenceInterne, RéférenceExterne
Types de messages reçus	ListeDesConversations, CréationDeConversation, Message, SuppressionDeBranche, LienAssociatif, RéférenceInterne, RéférenceExterne
Attributs	salleVirtuelle

AGENT DE ROLE DE CONFERENCE (ARC)	
identité, adresse	gérées par la Salle Virtuelle
Types de messages envoyés	MessagesNonLus, MESSagesNonValidés, ListeDesConversations, Conversation, Message, SuppressionDeBranche, ArbreDeConversation, LienAssociatif, RéférenceInterne, RéférenceExterne
Types de messages reçus	MessagesNonLus, MESSagesNonValidés, ListeDesConversations, Conversation, Message, ArbreDeConversation, LienAssociatif, RéférenceInterne, RéférenceExterne
Attributs	ConversationCourante

(2) Le composant Messages

Les types de messages échangés entre les instances des rôles de l'activité sont :

ListeDesConversations		
Etat	DEMANDE	RESULTAT
Requête	RESULTAT	
Expéditeur	PARTICIPANT	AGC
Destinataire	AGC	A
IDTâche	Identifiant de la tâche dont on veut connaître la liste des conversations	Identifiant de la tâche auxquelles appartiennent les conversations listées
Liste	-----	résultat de la requête : liste de conversations classées chronologiquement. Pour chaque conversation l'AGC fournit : son identifiant, son type, son titre, sa date de création, son auteur

Conversation		
Etat	CREATION	INFORMATION
Requête	INFORMATION	
Expéditeur	PARTICIPANT	AGC
Destinataire	AGC	ARC
IDTâche	Identifiant de la tâche dans laquelle le participant demande la création de la conversation	Identifiant local à la Salle Virtuelle de la tâche dans laquelle la conversation a été créée
IDConversation	-----	Identifiant local à la Salle Virtuelle de la Conversation décrite
Type	Type de conversation à instancier	Type de Conversation instancié
Titre	Titre de la conversation	Titre de la conversation
Date de création	-----	Date et heure à laquelle l'échange a été créé
Auteur	(expéditeur)	Identifiant de l'auteur

Message				
Etat	CREATION	MODIFICATION	VALIDATION	INFORMATION
Requête	INFORME	INFORME	DIFFUSE	
Expéditeur	PARTICIPANT	PARTICIPANT	PARTICIPANT	AGC
Destinataire	AGC	AGC	AGC	PARTICIPANT
MessagePère	Identifiant du message auquel le message répond	-----	-----	Identifiant du message auquel le message décrit répond
Description	Description du message à créer, sans identifiant	Description du message à modifier, avec identifiant	Identifiant du message à valider	Description du message, avec identifiant

LienAssociatif			
Etat	CREATION	SUPPRESSION	CONFIRMATION
Requête	CONFIRMATION	INFORMATION	
Expéditeur	ARC	ARC	AGC
Destinataire	AGC	AGC	ARC
Source	Identifiant du message source du lien	Identifiant du message source du lien	Identifiant du message source du lien
Cible	Identifiant du message cible du lien	Identifiant du message cible du lien	Identifiant du message cible du lien

RéférenceInterne			
Etat	CREATION	SUPPRESSION	CONFIRMATION
Requête	CONFIRMATION	CONFIRMATION ³	
Expéditeur	ARC	ARC	AGC
Destinataire	AGC	AGC	ARC
IDMessage	Identifiant du message auquel la référence doit être attachée	Identifiant du message auquel la référence est attachée	Identifiant du message auquel la référence est attachée
IDRéférence	Identifiant de l'ancre	Identifiant de l'ancre	Identifiant de l'ancre

RéférenceExterne			
Etat	CREATION	SUPPRESSION	CONFIRMATION
Requête	CONFIRMATION	CONFIRMATION ⁴	
Expéditeur	ARC	ARC	AGC
Destinataire	AGC	AGC	ARC
IDMessage	Identifiant du message auquel la référence doit être attachée	Identifiant du message auquel la référence est attachée	Identifiant du message auquel la référence est attachée
Description	Description de l'information référencée	Description de l'information référencée	Description de l'information référencée

SuppressionDeBranche	
Etat	DEMANDE
Requête	CONFIRME
Expéditeur	PARTICIPANT
Destinataire	AGC
MessageID	Identifiant du message racine de la branche à détruire

L'AGC confirme par l'envoi d'un message *ArbreDeConversation*.

³ L'AGC confirme la suppression d'une référence attachée à un message par l'envoi de la description du message.

⁴ L'AGC confirme la suppression d'une référence attachée à un message par l'envoi de la description du message.

(3) Le composant Fonctions

CREER une Conversation DANS une Tâche

Exécutée par l'AGC pour créer et enregistrer une activité de conversation dans la Tâche. La Salle Virtuelle affecte à la conversation un identifiant d'objet partagé qui permettra de la désigner ultérieurement. Un Agent de Médiation associé à la Conversation est créé. Le rôle de l'initiateur de la conversation est instancié et initialisé. La conversation sera validée et diffusée et les rôles des autres participants seront créés lorsque l'initiateur de la conversation validera le premier message de la conversation.

CREER un Lien DANS une Salle Virtuelle

Exécutée par l'AGC pour enregistrer un lien qui relie deux messages dans la salle virtuelle où se déroule la conférence.

CREER une Référence Interne DANS une Salle Virtuelle

Exécutée par l'AGC pour enregistrer une référence d'un message à une partie d'un document hypermédia de la librairie.

CREER une Référence Externe DANS une Salle Virtuelle

Exécutée par l'AGC pour enregistrer une référence d'un message à un objet externe au système informatique.

CREER un Message DANS une Salle Virtuelle

Exécutée par l'AGC pour enregistrer le message dans la Salle Virtuelle où se déroule la conférence. La Salle Virtuelle affecte au message un identifiant d'objet partagé qui permettra de le désigner ultérieurement (entre autres pour y répondre). L'AGC crée le lien de réponse entre le nouveau message et son père. Si le message est le premier d'une conversation, il est enregistré comme message racine de cette conversation. L'Agent de Médiation de la Conversation concerné est averti de la création du message et gère l'évolution de la conversation et des rôles.

SUPPRIMER un Message DANS une Salle Virtuelle

Exécutée par l'AGC pour supprimer un message et ses descendants. Supprime aussi tous les liens impliquant ce message dans la Salle Virtuelle.

VALIDER un Message

Exécutée par l'AGC pour valider le message. L'Agent de Médiation de la Conversation est averti de la validation du message et gère l'évolution de la conversation et des rôles impliqués dans la conversation.

CONVERSATIONS DANS une tâche POUR un participant

Exécutée par l'AGC pour déterminer la liste des conversations accessibles au participant dans la tâche. Pour chaque conversation dans la tâche, l'AGC demande à l'Agent de Médiation si elle est accessible au participant.

(4) Le composant Règles

Suite à l'analyse par la méthode AMIGO d'une activité de "Bulletin Board", [PAYS 89] explique que les règles sont simples mais que la complexité de l'activité (Gestion, stockage et parcours du graphe des messages et des liens) est encapsulée dans les fonctions. Nous parvenons à la même conclusion pour notre activité de conférence. Il n'y a pas de notion d'état global de l'activité et donc pas de phases.

```

A RECEPTION D'UNE ListeDesConversations EN demande
  laTâche = TROUVER ListeDesConversations.IDTâche DANS salleVirtuelle
  liste = CONVERSATIONS_DANS laTâche LISIBLE PAR expéditeur
  ENVOYER liste A expéditeur
A RECEPTION D'UNE Conversation EN création
  laConversation = CREER Conversation DANS: SalleVirtuelle
  ENVOYER laConversation A expéditeur
A RECEPTION D'UN Message EN création
  leMessage = CREER Message DANS salleVirtuelle
  ENVOYER leMessage A expéditeur
A RECEPTION D'UN Message EN modification
  leMessage = MODIFIER Message DANS salleVirtuelle
  ENVOYER CONVERSATION DE leMessage A expéditeur
  ENVOYER leMessage A expéditeur
A RECEPTION D'UNE Validation
  laTâche = TROUVER Validation.IDTâche DANS salleVirtuelle
  POUR CHAQUE message PARMIS LES (MESSAGES A VALIDER PAR expéditeur DANS
laTâche)
    VALIDER message
    laConversation = CONVERSATION DE message
    POUR CHAQUE participant PARMIS LES (LECTEURS DE laTâche)
      SI laConversation LISIBLE PAR participant
        ENVOYER laConversation A participant
    POUR CHAQUE participant PARMIS LES (LECTEURS DE laConversation)
      ENVOYER ARBRE DE laConversation A participant
    POUR CHAQUE lien PARMIS LES (LIENS VERS message)
      messageLié = SOURCE DE lien
      conversationLiée = CONVERSATION DE messageLié
      POUR CHAQUE participant PARMIS LES (LECTEURS DE
conversationLiée)
        SI lien LISIBLE PAR participant
          ENVOYER lien A participant
    POUR CHAQUE lien PARMIS LES (LIENS DE message)
      messageLié = CIBLE DE lien
      conversationLiée = CONVERSATION DE messageLié
      POUR CHAQUE participant PARMIS LES (LECTEURS DE
conversationLiée)
        SI lien LISIBLE PAR participant
          ENVOYER lien A participant

```

```

A RECEPTION D'UNE SuppressionDeBranche EN demande
  leMessage = TROUVER SuppressionDeBranche.IDMessage DANS salleVirtuelle
  laConversation = CONVERSATION DE leMessage
  SUPPRIMER leMessage DANS salleVirtuelle
  SI leMessage == RACINE DE laConversation
    laTâche = TACHE DE laConversation
    liste = CONVERSATIONS DANS laTâche LISIBLE PAR expéditeur
    ENVOYER liste A expéditeur
  SINON
    ENVOYER laConversation A expéditeur
A RECEPTION D'UN LienAssociatif EN création
  lien = CREER LienAssociatif DANS salleVirtuelle
  ENVOYER lien A expéditeur
A RECEPTION D'UNE RéférenceInterne EN création
  ref = CREER RéférenceInterne DANS salleVirtuelle
  ENVOYER ref A expéditeur
A RECEPTION D'UNE RéférenceExterne EN création
  ref = CREER RéférenceExterne DANS salleVirtuelle
  ENVOYER ref A expéditeur

```

(5) Le composant Droits

```

AUTORISE Conversation, Validation, Commentaire
AUTORISE_LECTURE Commentaire
INTERDIT_LECTURE message TEL QUE (message nonValidé ET (message auteur != participant))
INCLURE DROITS DE (ROLE DANS (CONVERSATION DE messageCourant))
SI messageCourant.auteur == participant ET ¬messageCourant.validé
  AUTORISE LienAssociatif, RéférenceInterne, RéférenceExterne, SuppressionDeBranche

```

b) Conception d'une conversation : l'exemple de la Conversation pour l'Action

Cette conversation a pour objet de négocier les conditions d'exécution d'une action. Actuellement, elle n'est conçue que pour supporter une négociation bilatérale. Cependant, du fait de son intégration dans la conférence, les autres membres du groupe peuvent être tenus au courant de cette négociation et peuvent l'influencer par des messages de types commentaire ou y faire référence dans des messages d'autres conversations. De plus, il sera facile de dériver cette classe d'activité pour concevoir une négociation multipartite, telle qu'un appel d'offre publique.

(1) Le composant Rôles

Agent de Médiation de Conversation pour l'Action (AMCA)	
identité, adresse	gérées par la Salle Virtuelle
Types de messages envoyés	Chgt_état, Chgt_phase, Chgt_règles
Types de messages reçus	Demande, Proposition, Accord, Report, Abandon, Refus, CompteRendu
Attributs	échangeCourant

Les Agents de Médiation de Conversation envoient des messages de médiation purs (changement d'état, changement de phase, changement de règles). La rétroaction de la modification des données de la conversation (création, suppression ou modification de messages, de liens, de références) est prise en charge par l'Agent de Gestion de Conférence.

DEMANDEUR	
identité, adresse	gérées par la Salle Virtuelle
Types de messages envoyés	Demande, Accord, Report, Abandon
Types de messages lus	Demande, Proposition, Accord, CompteRendu, Report, Refus, Abandon
Attributs	échangeCourant

SOLLICITE	
identité, adresse	gérées par la Salle Virtuelle
Types de messages envoyés	Proposition, Accord, Report, Refus, CompteRendu
Types de messages lus	Demande, Proposition, Accord, CompteRendu, Report, Refus, Abandon
Attributs	échangeCourant

(2) Le composant Messages

Demande		
Etat	CREATION	VALIDATION
Requête	MEMORISER	PUBLIER
Expéditeur	PARTICIPANT	PARTICIPANT
Destinataire	AMCA	AMCA
MessagePère	Identifiant du message auquel le message créé répond	-----
ID	-----	Identifiant du message à valider
titre	titre du message	-----
contenu	description de l'action souhaitée	-----
son	message vocal	-----
dateButoir	date limite pour la réalisation de l'action souhaitée	-----
sollicité	identifiant de la personne sollicitée pour l'action	-----

Proposition		
Etat	CREATION	VALIDATION
Requête	MEMORISER	PUBLIER
Expéditeur	PARTICIPANT	PARTICIPANT
Destinataire	AMCA	AMCA
MessagePère	Identifiant du message auquel le message répond	-----
ID	-----	Identifiant du message à valider
titre	titre du message	-----
contenu	description de l'action proposée	-----
son	message vocal	-----
dateButoir	date limite pour la réalisation de l'action proposée	-----

Refus		
Etat	CREATION	VALIDATION
Requête	MEMORISER	PUBLIER
Expéditeur	PARTICIPANT	PARTICIPANT
Destinataire	AMCA	AMCA
MessagePère	Identifiant du message auquel le message répond	-----
ID	-----	Identifiant du message à valider
titre	explication du refus	-----
son	message vocal	-----

Abandon		
Etat	CREATION	VALIDATION
Requête	MEMORISER	PUBLIER
Expéditeur	PARTICIPANT	PARTICIPANT
Destinataire	AMCA	AMCA
MessagePère	Identifiant du message auquel le message répond	-----
ID	-----	Identifiant du message à valider
titre	explication de l'abandon	-----
son	message vocal	-----

Accord		
Etat	CREATION	VALIDATION
Requête	MEMORISER	PUBLIER
Expéditeur	PARTICIPANT	PARTICIPANT
Destinataire	AMCA	AMCA
MessagePère	Identifiant du message auquel le message répond	-----
ID	-----	Identifiant du message à valider
titre	contenu textuel	-----
son	message vocal	-----

CompteRendu		
Etat	CREATION	VALIDATION
Requête	MEMORISER	PUBLIER
Expéditeur	PARTICIPANT	PARTICIPANT
Destinataire	AMCA	AMCA
MessagePère	Identifiant du message auquel le message répond	-----
ID	-----	Identifiant du message à valider
titre	contenu textuel	-----
son	message vocal	-----

Report		
Etat	CREATION	VALIDATION
Requête	MEMORISER	PUBLIER
Expéditeur	PARTICIPANT	PARTICIPANT
Destinataire	AMCA	AMCA
MessagePère	Identifiant du message auquel le message répond	-----
ID	-----	Identifiant du message à valider
titre	contenu textuel	-----
son	message vocal	-----
dateButoir	nouvelle date limite pour la réalisation de l'action	-----

(3) Le composant Fonctions

AFFECTER LE RÔLE *un Rôle A un Participant*

Exécutée par l'ACA pour attribuer un rôle à un participant dans la conversation. Le rôle définit les règles de conversation d'un participant. Ces règles déterminent les types de messages que le participant peut générer pour répondre à un message. Les règles de conversation permettent également de sélectionner les messages de la conversation qui peuvent être inclus dans la vue du participant sur la conversation.

MAJ ACTION SELON *un Message*

Exécutée par l'ACA pour mettre à jour le noyau fonctionnel de l'activité.

DERNIERE PROPOSITION

Exécutée par l'Agent de Rôle pour trouver dans la conversation la dernière proposition.

(4) Le composant Règles

Comme le montre la figure 41, la conversation pour l'action que nous avons conçue comprend huit phases : initialisée, demandée, acceptée, reportée, refusée, abandonnée et terminée.

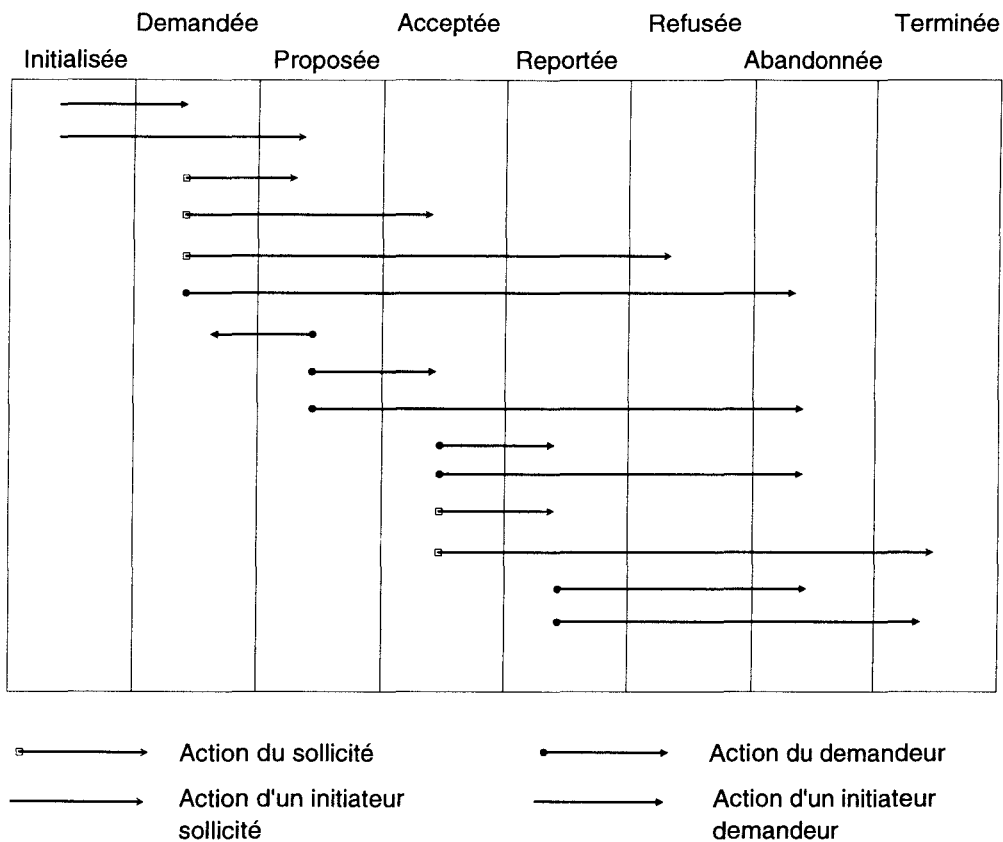


Figure 41. L'évolution d'une Conversation pour l'Action

DEBUT DE PHASE Initialisée

- A RECEPTION D'UNE Demande créée
 - demandeur = Demande.Expéditeur
 - sollicité = Demande.Sollicité
 - AFFECTER LE RÔLE Demandeur A demandeur
 - PASSER demandeur DANS L'ETAT doitValider
- A RECEPTION D'UNE Demande validée DE demandeur
 - AFFECTE LE RÔLE sollicité A sollicité
 - MAJ ACTION SELON Demande
 - PASSER EN PHASE Demandée
- A RECEPTION D'UNE Proposition créée FAIRE
 - sollicité = Demande.Expéditeur
 - demandeur = RESPONSABLE DE tâche
 - AFFECTE LE RÔLE Sollicité A sollicité
 - PASSER sollicité DANS L'ETAT doitValider
- A RECEPTION D'UNE Proposition validée DE sollicité
 - MAJ ACTION SELON Proposition
 - AFFECTER LE RÔLE Demandeur A demandeur
 - PASSER EN PHASE Négociée

FIN DE PHASE

DEBUT DE PHASE Demandée

- PASSER sollicité DANS L'ETAT neutre
- PASSER demandeur DANS L'ETAT neutre
- A RECEPTION D'UN Message créé
 - PASSER expéditeur DANS L'ETAT doitValider

PASSER AUTRE QUE expéditeur DANS L'ETAT attente
A RECEPTION D'UNE Proposition validée DE sollicité
MAJ ACTION SELON Proposition
PASSER EN PHASE Négociée
A RECEPTION D'UN Accord validé DE sollicité
PASSER EN PHASE Acceptée
A RECEPTION D'UN Refus validé DE sollicité
PASSER EN PHASE Refusée
A RECEPTION D'UN Abandon validé DE demandeur
PASSER EN PHASE Abandonnée
FIN DE PHASE

DEBUT DE PHASE Négociée
PASSER sollicité DANS L'ETAT neutre
PASSER demandeur DANS L'ETAT neutre
A RECEPTION D'UN Message créé
PASSER expéditeur DANS L'ETAT doitValider
PASSER AUTRE QUE expéditeur DANS L'ETAT attente
A RECEPTION D'UNE Demande validée DE demandeur
MAJ ACTION SELON Proposition
PASSER EN PHASE Demandée
A RECEPTION D'UN Accord validé DE demandeur
PASSER EN PHASE Acceptée
A RECEPTION D'UN Abandon validé DE demandeur
PASSER EN PHASE Abandonnée
FIN DE PHASE

DEBUT DE PHASE Acceptée
PASSER sollicité DANS L'ETAT neutre
PASSER demandeur DANS L'ETAT neutre
A RECEPTION D'UN Message créé
PASSER expéditeur DANS L'ETAT doitValider
PASSER AUTRE QUE expéditeur DANS L'ETAT attente
A RECEPTION D'UN Abandon validé DE demandeur
PASSER EN PHASE Abandonnée
A RECEPTION D'UN Compte-Rendu validé DE sollicité
PASSER EN PHASE Terminée
A RECEPTION D'UN Report validé
MAJ ACTION SELON Report
PASSER EN PHASE Reportée
FIN DE PHASE

DEBUT DE PHASE Reportée
PASSER sollicité DANS L'ETAT neutre
PASSER demandeur DANS L'ETAT neutre
A RECEPTION D'UN Message créé
PASSER expéditeur DANS L'ETAT doitValider
PASSER AUTRE QUE expéditeur DANS L'ETAT attente
A RECEPTION D'UN Message créé
PASSER expéditeur DANS L'ETAT doitValider
A RECEPTION D'UN Abandon validé DE demandeur
PASSER EN PHASE Abandonnée
A RECEPTION D'UN Compte-Rendu validée DE sollicité
PASSER demandeur DANS L'ETAT aValidé
PASSER EN PHASE Terminée

A RECEPTION D'UN Report validé
 MAJ ACTION SELON Report
 PASSER EN PHASE Reportée
 FIN DE PHASE

DEBUT DE PHASE Abandonnée
 STOPPE ACTIVITE
 FIN DE PHASE

DEBUT DE PHASE Refusée
 STOPPE ACTIVITE
 FIN DE PHASE

DEBUT DE PHASE Terminée
 STOPPE ACTIVITE
 FIN DE PHASE

(5) Le composant Droits

Les règles de conversation d'une activité de Conversation en temps différé dans une conférence sont plus complexes que celles d'une activité de communication de groupe en temps réel. Pour déterminer l'ensemble des types de messages qu'un rôle peut émettre il ne suffit pas de considérer l'état du rôle et la phase de l'activité : il faut aussi tenir compte du message auquel le participant veut répondre. Dans les activités les plus simples (vote, tour de table), le problème peut être simplifié en se limitant à tenir compte du type du message. Mais lorsque la Conversation contient des paires récurrentes, comme dans le cas d'une négociation, les droits de réponse changent d'une instance à l'autre d'un même type de message dans une même conversation. Par exemple, dans le cas d'une négociation, un participant peut répondre à la dernière proposition de son interlocuteur mais pas aux propositions passées. Cette complexité nécessite l'exécution par l'Agent de rôle de fonctions spécifiques à l'activité, telle que la fonction DERNIERE PROPOSITION que nous utilisons ici.

Les règles de conversation suivies par un initiateur⁵

AUTORISE_LECTURE Demande, Proposition

Selon (Etat,Phase)

(neutre, Init) :

SI messageCourant == NULL

AUTORISE Demande, Proposition, Compte-Rendu

Les règles de conversation suivies par un demandeur

AUTORISE_LECTURE Demande, Proposition, Accord, Report, Refus, Abandon, Compte-Rendu

Selon (Etat,Phase)

(neutre, Négociée) :

SI messageCourant == DERNIERE PROPOSITION

⁵ Le rôle d'initiateur est un rôle transitoire donné au créateur d'une conversation pour l'action. Le premier message qu'il crée dans la conversation permet de déterminer le rôle qu'il jouera dans la conversation (demandeur ou sollicité)

AUTORISE Accord, Demande, Abandon
 (neutre, Acceptée) :
 SI TYPE DE messageCourant == Accord
 AUTORISE Abandon, Report
 (neutre, Reportée) :
 SI TYPE DE messageCourant == Report
 AUTORISE Abandon

Les règles de conversation suivies par un sollicité

AUTORISE_LECTURE Demande, Proposition, Accord, Report, Refus, Abandon, Compte-Rendu
 Selon (Etat,Phase)

(neutre, Demandée) :
 SI messageCourant == DERNIERE PROPOSITION
 AUTORISE Accord, Proposition, Refus
 (neutre, Acceptée) :
 SI TYPE DE messageCourant == Accord
 AUTORISE Compte-Rendu, Report
 (neutre, Reportée) :
 SI TYPE DE messageCourant == Report
 AUTORISE Report, Compte-Rendu

c) Conclusion

Le modèle Co-Learn enrichit le modèle AMIGO pour permettre d'analyser et concevoir des **Activités de Conversation de Groupe** en temps différé dans le cadre d'une **Conférence**. Contrairement aux approches habituelles, nous privilégions la communication de groupe par rapport à la communication interpersonnelle.

Par contre, nous concilions les avantages de l'approche classique adoptée par AMIGO de centralisation de la coordination sur un médiateur, avec l'approche de distribution entre les **Rôles**, grâce à l'ajout d'un composant **Droits**.

La gestion des droits d'un **Rôle** dans une Activité de Conversation de Groupe en Temps Différé définit les droits d'accès à deux types de ressources : les **Ressources d'Information** et les **Ressources d'Action**.

Le contrôle des Ressources d'Information consiste à filtrer les messages de conversation à inclure dans la **Vue de la Conversation** que le système doit présenter au rôle. Le contrôle des Ressources d'Action consiste à conseiller l'utilisateur dans l'utilisation des **Types de Réponses**. Ces deux tâches de contrôles nécessitant de nombreux accès à la base de messages, il est préférable de les confier aux **Objets de Rôle sur les serveurs** coopérant avec les **Objets de Médiation** et les **Objets Gestion de la Conférence**.

1.2.3. Conclusion sur la conception des activités de coopération

Nous ajoutons un **composant Droits** pour analyser et concevoir les activités en temps réel, comme les activités en temps différé.

Le rôle de ce composant Droits est le même dans les deux cas (temps réel et temps différé) : contrôler l'accès des rôles aux Ressources d'Information et aux Ressources d'Action.

Les droits de lecture des messages de conversation dans les activités asynchrones de conférence et de conversation sont l'alter ego des **Droits sur les Ressources d'Information** des activités de coopération en temps réel.

Dans le cas des Activités en temps réel, le contrôle de ces droits est **délégué** aux **Agents de Rôle** distribués qui contrôlent les **Agents d'Interface**. Cette délégation nécessite la définition d'un protocole qui se traduit dans le modèle par les messages génériques Chargement de Règles, Changement de Phase et Changement d'Etat.

Dans le cas des activités de communication en temps différé, il est préférable de ne pas déléguer la gestion de ces droits à des Agents de Rôles distribués auprès des Agents d'Interface. Il est à la fois plus simple et plus efficace de confier le contrôle des droits à des **Objets de Rôles** qui coopèrent avec les **Objets Médiateurs** et les **Agents de Gestion de Conférence** sur les serveurs pour construire les **Vues** adaptées aux rôles. Dès lors, une réelle coopération objet remplace le protocole de délégation du temps réel. Les objets médiateurs étant en relation rapide et permanente avec les objets de rôles dans un même espace d'objets, le protocole de communication n'a pas besoin d'être allégé. De ce fait, les objets rôles accèdent rapidement à toute l'information nécessaire pour prendre en charge l'évaluation de leur état dont se trouvent déchargés les objets médiateurs.

A notre connaissance, c'est que l'on voit concilier par une même approche la conception des applications de travail coopératif synchrones et asynchrones.

2. LE DEUXIEME NIVEAU DE LA PYRAMIDE : LA GESTION DES UNITES ORGANISATIONNELLES

La Salle Virtuelle permet de délimiter un environnement de coopération. L'ensemble des Salles Virtuelles forme un espace global qu'il s'agit de structurer.

Nous nous intéressons donc au deuxième étage de la pyramide de l'organisation. Au premier étage les personnes se réunissent dans des salles virtuelles pour former des équipes de travail. Au deuxième niveau, les personnes sont regroupées dans des unités organisationnelles : départements, laboratoires, bureaux d'étude, divisions, modules de formation...

Nous n'excluons pas qu'un groupe organisationnel puisse correspondre à une seule équipe, ni que certaines équipes soient composées de membres de plusieurs groupes organisationnels. Pour supporter ceci, nous permettons que les groupes organisationnels ne soient pas disjoints.

Un nouveau niveau de malléabilité fonctionnel est offert en permettant d'affecter les rôles organisationnels aux membres du groupe. Actuellement, le système offre trois rôles : professeur, tuteur et apprenant.

Un professeur est responsable du module de formation : il organise le module en nommant des tuteurs.

Un tuteur a pour rôle de créer les Salles Virtuelles dans lesquelles auront lieu les sessions d'apprentissage. Pour chaque Salle Virtuelle, il nomme les personnes auxquelles il délègue le rôle de tuteur (il peut bien sûr se nommer lui-même). Il peut en effet confier à une personne inscrite comme apprenant au module de formation le rôle de configurer les Salles Virtuelles et d'animer les sessions d'apprentissage qui s'y déroulent.

Les professeurs peuvent inscrire eux-mêmes les apprenants dans le module de formation dont ils sont responsables. Cependant, ils peuvent décider de laisser ce travail au personnel administratif, comme nous le verrons dans l'analyse de la gestion générale de l'organisation.

De même, les tuteurs du module de formation peuvent inscrire eux-mêmes les apprenants dans les Salles Virtuelles ou laisser cette tâche soit aux personnes auxquelles ils délèguent le rôle de tuteur dans les Salles Virtuelles soit au personnel administratif.

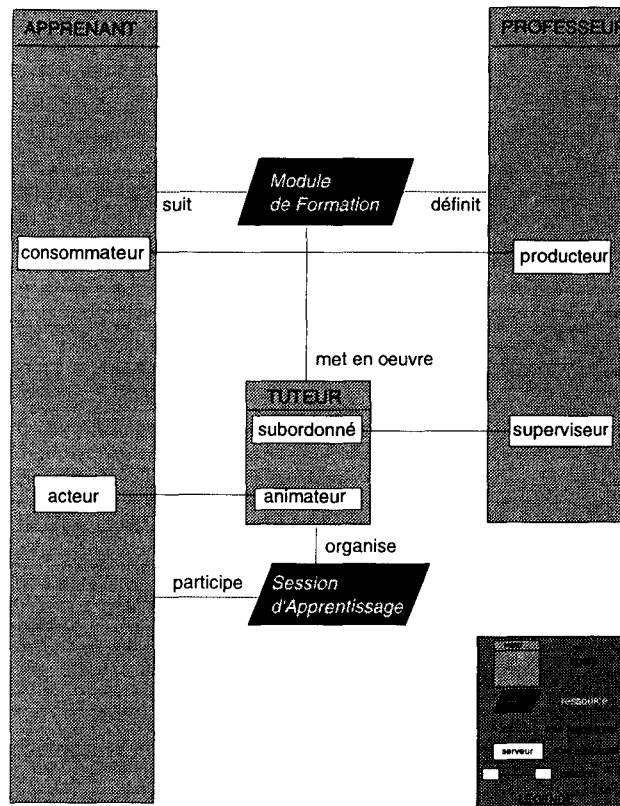


Figure 42. Modélisation de type ORDIT de l'organisation des modules de formation d'un institut d'enseignement

3. LE SOMMET DE LA PYRAMIDE : LA GESTION GENERALE DE L'ORGANISATION

Au niveau supérieur de la pyramide, nous trouvons les rôles impliqués dans l'organisation générale. Le système identifie trois rôles : l'administration du personnel, l'administration des structures (groupes organisationnels et salles virtuelles) et la participation aux situations de coopération.

Les personnes chargées de l'administration du personnel ont toujours la responsabilité d'enregistrer les membres de l'organisation au niveau global, parfois dans les modules de formation et plus rarement dans les salles virtuelles. Il s'agit essentiellement d'enregistrer les personnes responsables de l'administration des structures et les apprenants dans l'organisation. Souvent, le personnel administratif aura la responsabilité de l'inscription des stagiaires

L'administration des structures consiste à créer des structures (dans le cas d'un institut d'enseignement, des modules de formation) et y inscrire des membres avec leurs rôles.

Cet ensemble de rôles n'est pas très riche. Nous privilégions pour l'instant la simplicité de gestion des rôles organisationnels car le but premier du système n'est pas de supporter l'organisation mais d'assister les situations d'apprentissage.

Le schéma organisationnel a donc pour objet de fournir un contexte organisationnel à la situation d'apprentissage. La priorité est donc d'explicitier le statut des intervenants plutôt que de leur donner tous les moyens de remplir leurs fonctions. D'ailleurs, dans les expérimentations, le système n'est pas utilisé par les directeurs et les administratifs. Seuls les formateurs et les apprenants l'utilisent. Selon l'organisation, ce sont les formateurs ou l'informaticien administrateur du serveur Co-Learn qui jouent les rôles d'administration.

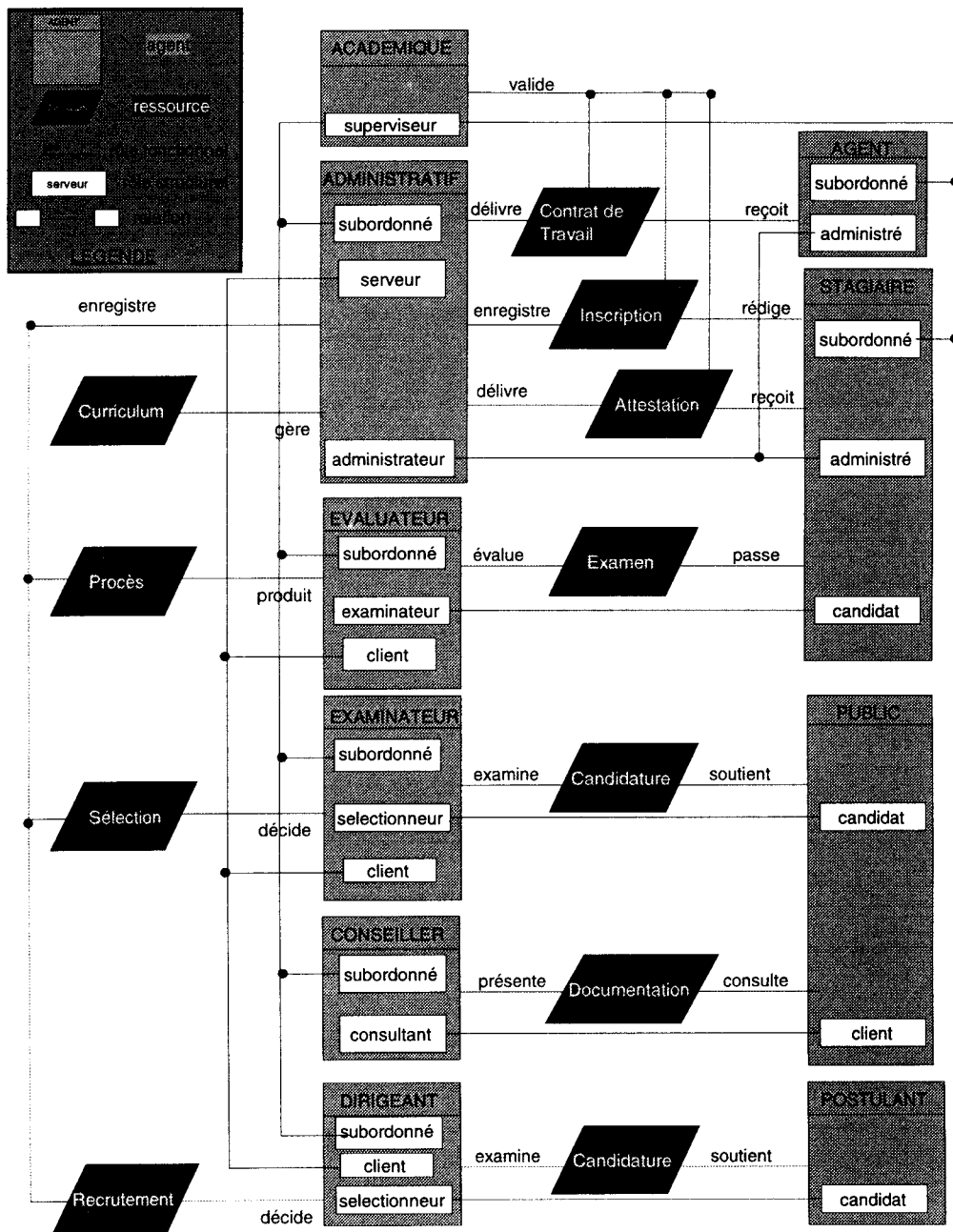


Figure 43. Modélisation de type ORBIT de l'organisation générale d'un institut de formation

Co-Learn propose l'outil d'exploration de l'organisation (figure 45) qui tient les utilisateurs au courant de la présence des participants dans une Salle Virtuelle. Cet outil permet également de connaître tous les rôles qu'un participant joue aux divers niveaux de l'organisation. Il est possible de connaître l'ensemble des groupes organisationnels et l'ensemble des salles virtuelles dont un participant fait partie. Cet outil, comme de nombreux outils Co-Learn est un outil actif : il ne se contente pas de répondre aux requêtes de l'utilisateur : il l'avertit également des modifications pertinentes de l'état de l'organisation (par exemple l'arrivée et le départ des participants dans une Salle Virtuelle). L'outil exploite les possibilités multimédia du poste Co-Learn pour informer l'utilisateur. Ainsi l'arrivée ou le départ d'un participant d'une Salle Virtuelle sont signalés par un message sonore.

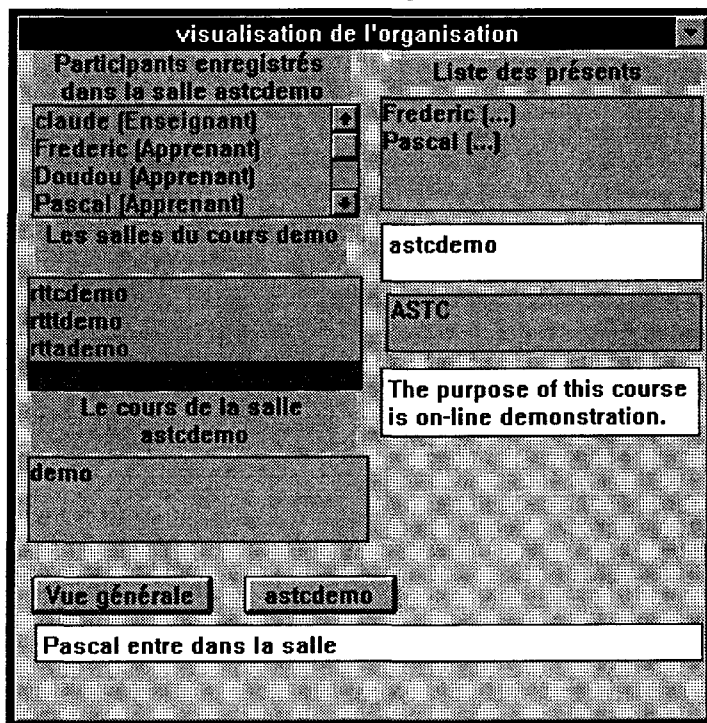


Figure 45. Interface de l'outil d'exploration de l'organisation

L'outil de gestion des actions (figure 46) participe aussi à donner des vues sur les participants. Il permet de connaître :

- l'ensemble des actions qu'un participant s'est engagé à mener,
- l'ensemble des actions promises à un participant,
- l'ensemble des actions réalisées par un participant,
- l'ensemble des actions que des participants sont en train de négocier.

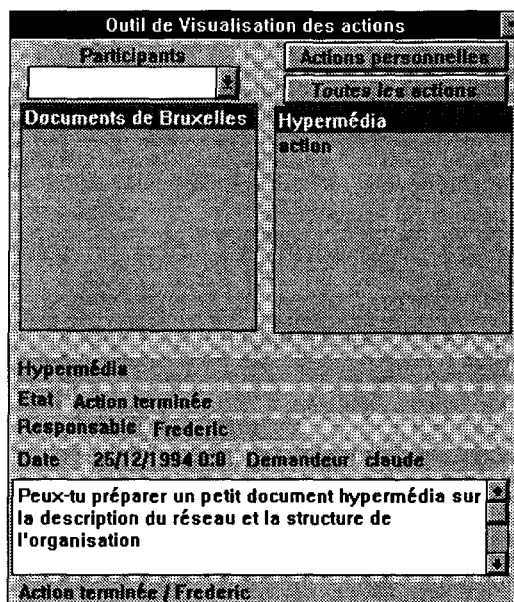


Figure 46. Interface de l'outil de gestion des actions

4.3. Vues sur le contexte

Co-Learn offre une vue du **contexte structurel**. Par exemple, l'outil d'aide à la conversation (figure 47) présente à un utilisateur l'ensemble des messages qu'il n'a pas encore lus et l'ensemble des messages et conversations qu'il a créés et pas encore publiés.

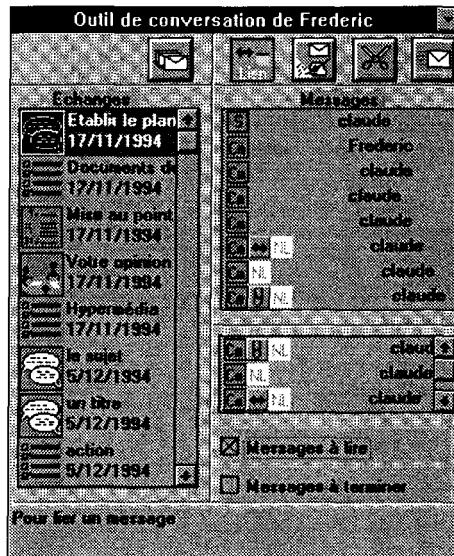


Figure 47. Interface de l'outil d'aide à la conversation

La boîte à outils privée présente les outils utilisés dans la Salle Virtuelle. L'outil de gestion des actions permet de connaître les actions du groupe dans lesquelles nous sommes engagés (figure 46). L'outil de gestion des rendez-vous (figure 48) nous donne aussi une vue du contexte structurel.

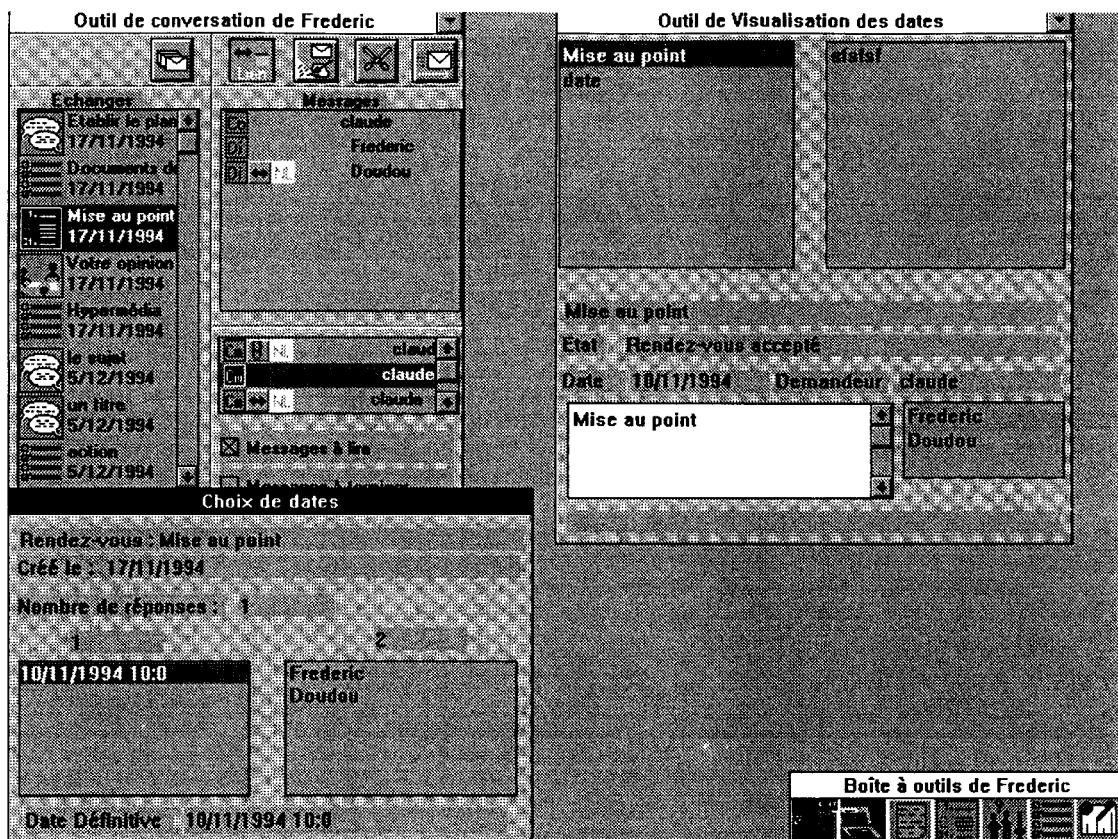


Figure 48. Interface de l'outil de gestion des rendez-vous du groupe

La situation de la Salle Virtuelle dans un groupe organisationnel participe au **contexte organisationnel** de la coopération. La procédure d'entrée dans une pièce renforce la connaissance de ce contexte. L'outil d'exploration de l'organisation le rappelle (Cf. figure 45).

CHAPITRE 11

LES POINTS DE VUE DE L'INFORMATION

Nous identifions deux points de vue sur l'information :

- Le point de vue sur l'information qu'ont les utilisateurs du système : c'est ce que [ELLIS 94] appelle le modèle ontologique.

- Le point de vue des concepteurs du système et des applications : c'est par exemple les points de vue sur l'information de MOCCA, ODP et X500 et le point de vue modélisé par le modèle conceptuel de données d'AMIGO MHS+.

L'utilisation de l'approche orientée objet permet d'obtenir un modèle de conception qui ne diffère que peu du modèle ontologique.

Les méthodes objets telles que OOA et OMT considèrent les deux points de vue. Le point de vue de l'utilisateur est considéré dans la phase d'analyse pour construire le coeur du modèle objet. Ce modèle est enrichi dans la phase de conception en considérant le point de vue du concepteur qui ajoute les fonctions utilitaires, les mécanismes génériques et assure l'adaptation aux contraintes du système informatique.

1. LE MODELE ONTOLOGIQUE

Nous séparons le modèle ontologique en trois domaines :

- le domaine organisationnel
- le domaine de la coopération en temps différé
- le domaine de la coopération en temps réel

1.1. le domaine organisationnel

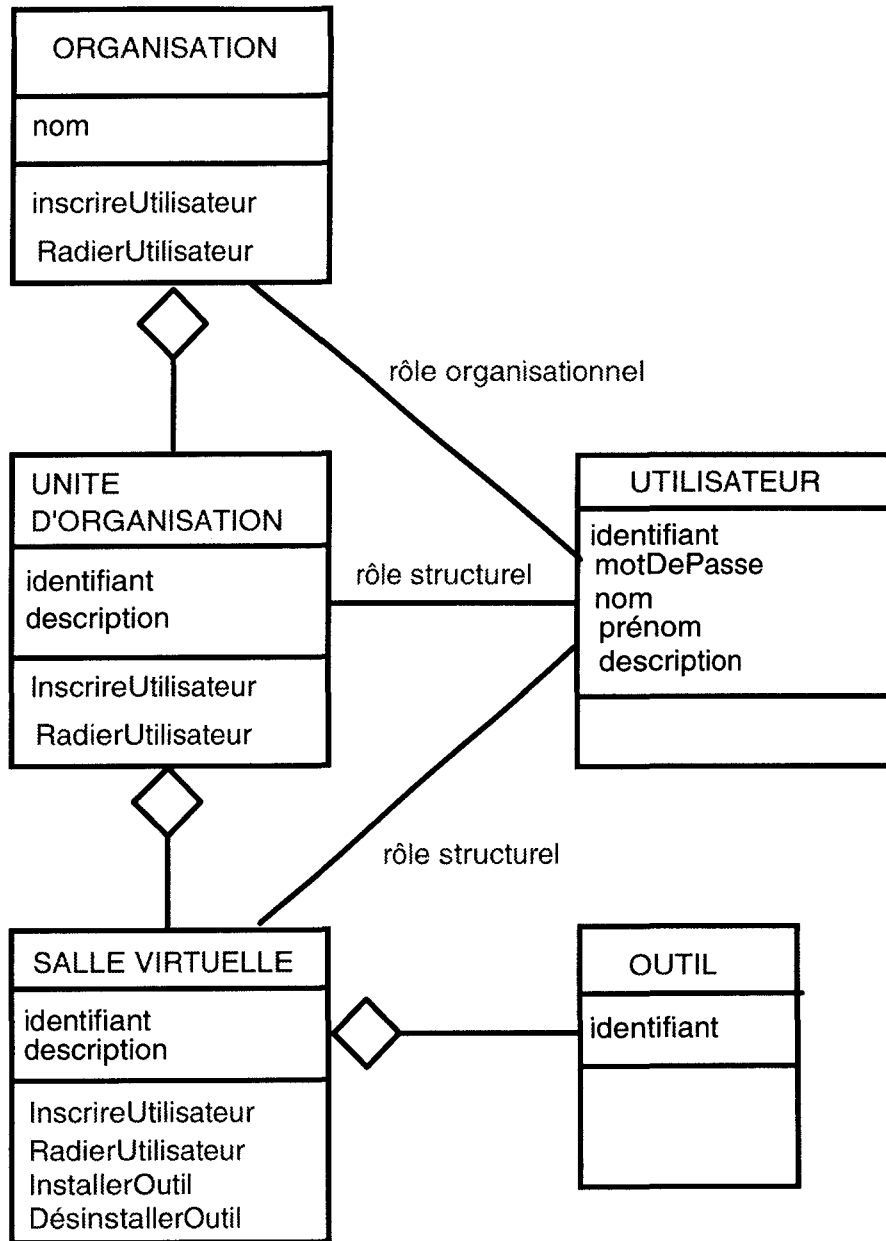


Figure 49. Modèle ontologique du domaine organisationnel

1.2. Le domaine de la coopération en temps différé

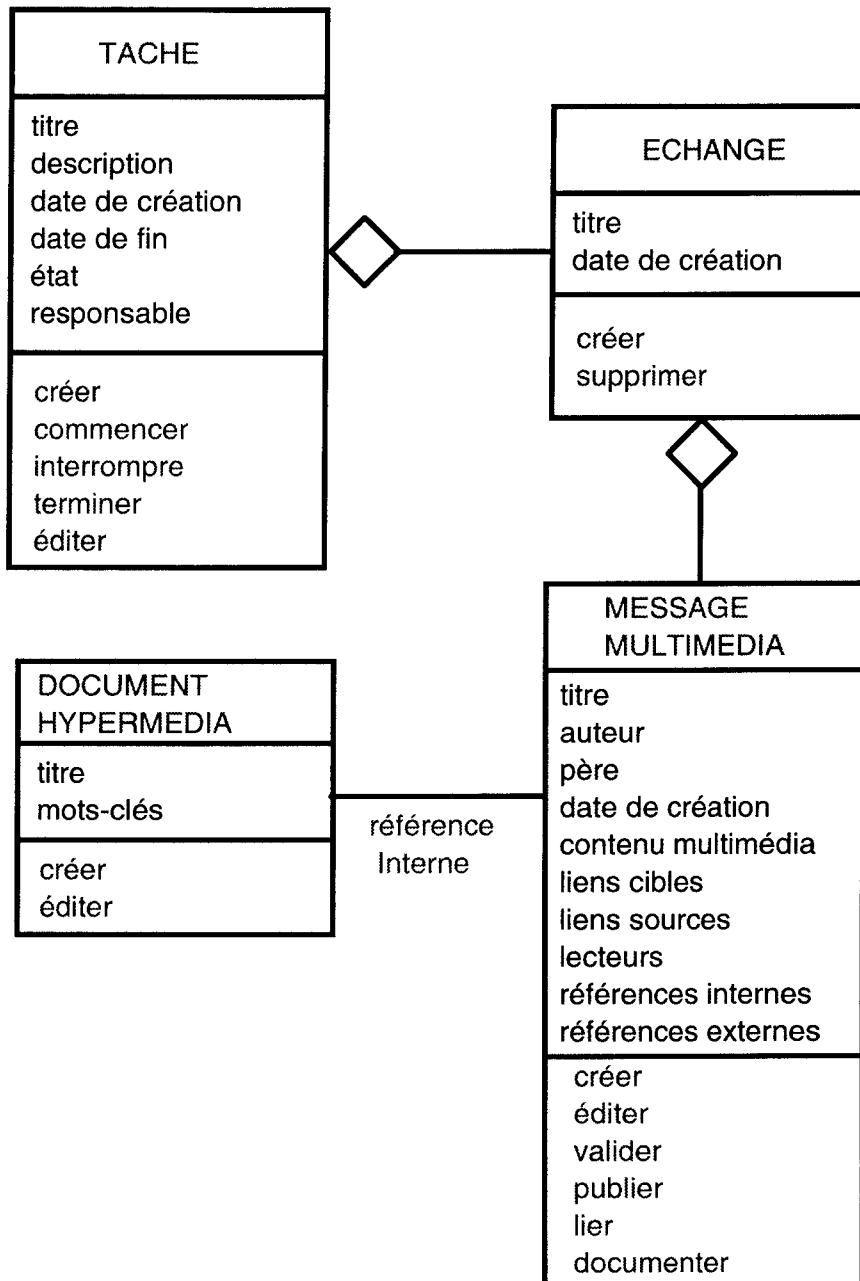


Figure 50. Le modèle ontologique de la coopération en temps différé

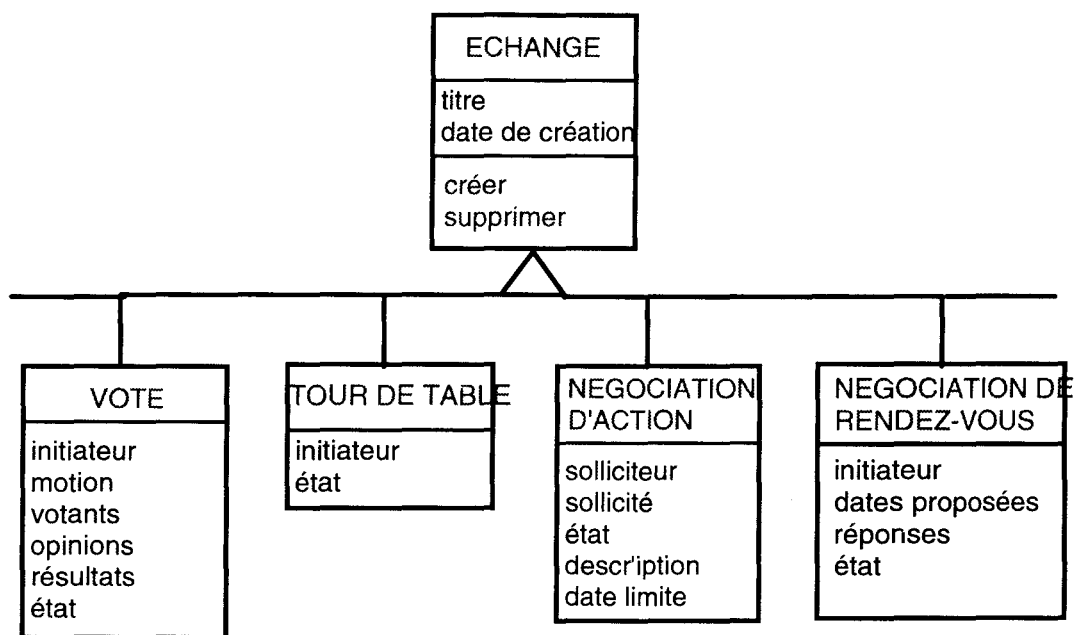


Figure 51. Le Modèle Ontologique de la Conversation

1.3. Le domaine de la documentation

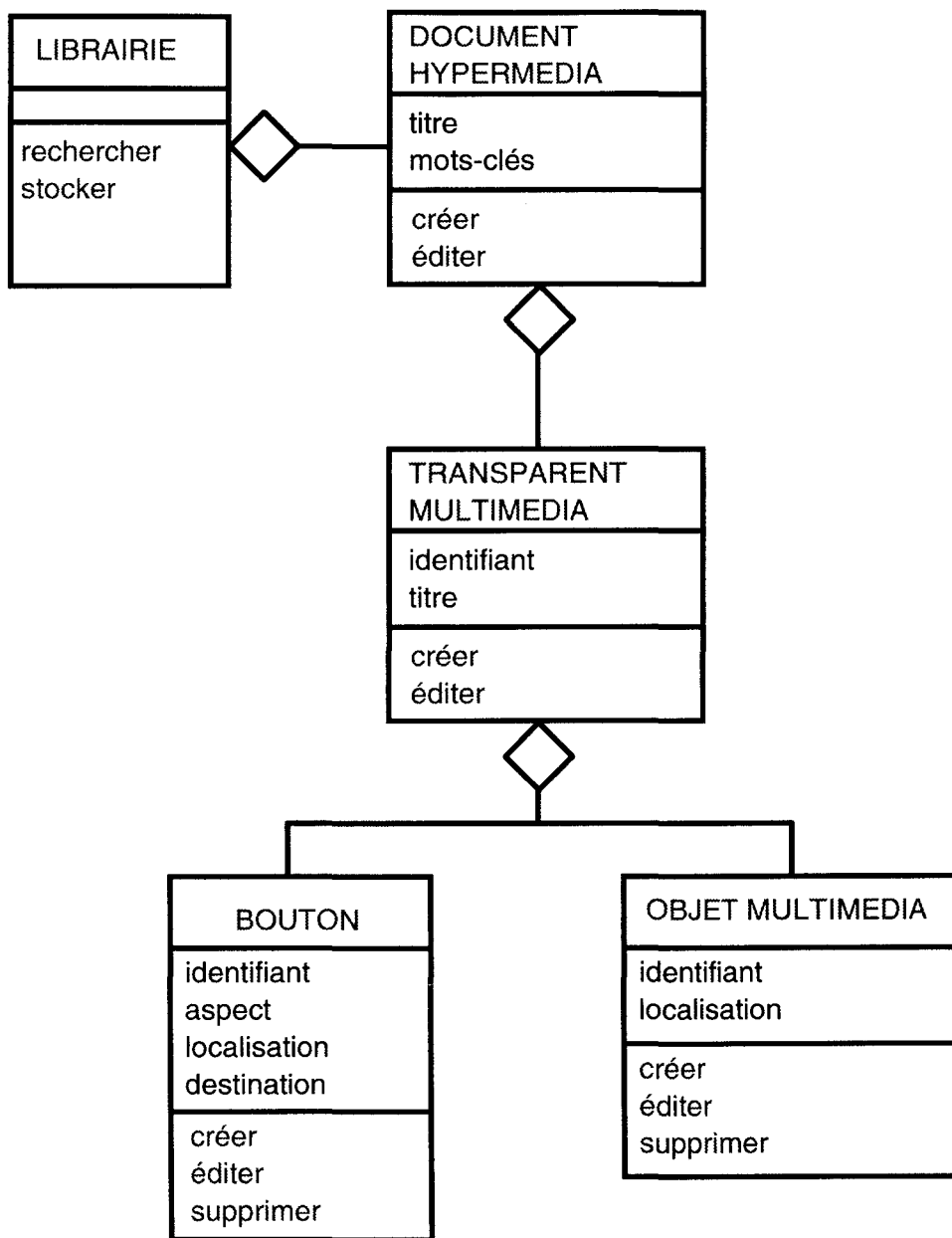


Figure 52. Le domaine de la documentation du modèle ontologique

1.4. Le domaine de la vue partagée

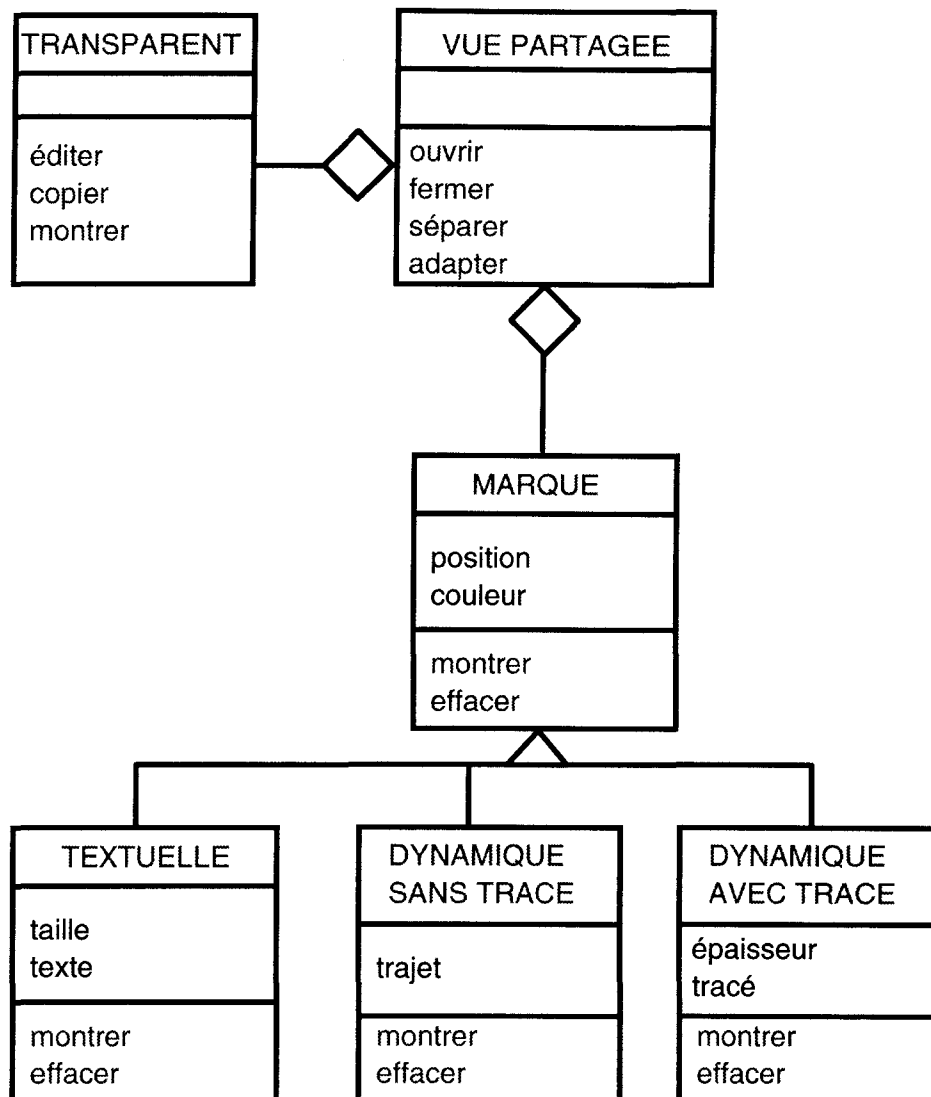


Figure 53. Le domaine de l'information partagée du modèle ontologique

2. LE MODELE CONCEPTUEL DES DONNEES

Il s'agit du point de vue du concepteur sur l'information. Lorsque le système d'information est réalisé à l'aide d'un environnement objet, ce modèle conceptuel définit le schéma de la base de données. Nous décrivons ici le schéma objet du serveur d'activités Co-Learn, que nous avons développé et qui est au coeur du système d'information de Co-Learn.

La particularité d'un système d'information d'aide au travail coopératif est qu'il intègre la connaissance de l'environnement dans lequel il est situé. Le système d'information utilise cette connaissance de deux façons. D'abord, il la restitue à bon escient aux utilisateurs du système qui, pour coopérer à distance ont besoin de connaître leur situation et celle de leurs collaborateurs. Mais surtout, il utilise cette connaissance pour s'adapter à l'organisation et aux utilisateurs. Par exemple, la connaissance des rôles des utilisateurs permet au système d'adapter son comportement afin de faciliter leur travail. Il utilise aussi la connaissance des rôles des utilisateurs et des règles de l'organisation pour représenter les utilisateurs auprès de leurs collaborateurs. En l'absence d'un utilisateur, l'environnement de travail coopératif observe l'évolution des activités de l'entreprise qui intéressent l'utilisateur absent et lui en rend compte lors de son retour.

Par conséquent, le modèle conceptuel et le système d'information de l'environnement d'aide au travail coopératif contiennent la connaissance relative à tous les points de vue utiles à l'analyse et la conception d'un tel environnement. C'est pourquoi nous proposons une analyse de notre système d'information cinq des points de vue identifiés par MOCCA. Cependant, nous étendons le point de vue de MOCCA sur la communication en considérant toutes les activités de groupe (de communication, de production ou de coordination) et en y incluant la description de la communication à l'intérieur de ces activités.

Il est important de noter que le système d'information objet que nous présentons ici forme un réseau presque complètement connecté. Ceci signifie que la totalité du système d'information peut être retrouvé à partir d'un seul de ses objets. Ceci reflète le fait que dans un environnement de travail coopératif, chaque élément participe à l'ensemble et a une position par rapport à tous les autres éléments. Chaque élément est à la fois sémantiquement pauvre voire inutile lorsqu'il est considéré isolé mais riche de sens et indispensable lorsqu'il est considéré dans ses relations et sa coopération avec les autres éléments.

2.1. Le domaine des Salles Virtuelles

Une **Salle Virtuelle** est un espace virtuel où des **Participants** interagissent entre eux et avec des **Outils**. Elle offre un éther permanent permettant la communication informelle à tout moment. Elle facilite la prise de conscience de l'existence des autres et de leurs activités. Le monde des Salles Virtuelles interconnectées permet la l'exploration de l'environnement de coopération et la localisation des personnes, des outils et des **Activités**. La structuration des Salles Virtuelles fait partie du modèle organisationnel. La Salle Virtuelle rend l'utilisation de tout outil collaboratif en rendant compte de cette utilisation aux autres participants, en permettant une communication informelle autour de cet outil et en situant cette utilisation dans un contexte social.

Dans un environnement d'enseignement ouvert, la métaphore de Salle Virtuelle est utilisée pour structurer l'espace des Activités d'Apprentissage Coopératifs [DERY 92] et les contextualiser. Les apprenants, les experts et les formateurs participent à des activités qui ont lieu dans les Salles Virtuelles. Ceci permet la construction sociale du savoir. La Salle Virtuelle est l'unité d'intégration d'un groupe de participants, d'un ensemble d'outils complémentaires utiles à la réalisation de la tâche commune.

Ainsi, les classes d'objets du système d'information qui sont au coeur du modèle de la Salle Virtuelle sont la Salle Virtuelle (Room), l'Activité (Activity), le Groupe de Travail (WorkGroup) et le Participant (Participant). Ce sont en fait les points d'entrée dans le système d'information privilégiés par le modèle de la Salle Virtuelle. Leur intérêt réside dans les services qu'ils rendent mais également dans la façon dont ils donnent accès aux autres points de vue.

2.1.1. La classe Salle Virtuelle

Une instance de Salle Virtuelle offre les services internes au modèle de la Salle Virtuelle :

- elle met en relation les participants et les activités :
elle communique aux activités les actions des participants,
elle communique aux participants les évolutions des activités,
- elle gère les entrées et sortie en session des participants

Elle constitue aussi un pont vers :

- le modèle Organisationnel par sa relation avec l'instance d'Organisation dont elle dépend,

- le modèle de l'Architecture Distribuée, parce que, dérivant de la classe Objet Connecté, elle coopère avec un objet de type Communicateur qui assure l'interface avec les services de communication distribués,
- le modèle de la Communication en offrant des services de communication entre les activités

2.1.2. La classe Participant

Un objet de type Participant offre des services internes au modèle de la Salle virtuelle :

- il gère l'implication des utilisateurs dans les activités de la Salle par sa connaissance de l'ensemble des rôles de l'utilisateur dans les activités de la Salle,

Il constitue un pont vers :

- le modèle de l'Expertise Utilisateur par sa coopération avec l'instance d'Utilisateur qu'il représente dans la Salle
- le modèle de la Communication par sa connaissance des rôles du Participant dans les activités

2.1.3. La classe Activité

Une instance d'Activité offre des services internes au modèle de la Salle Virtuelle :

- elle communique l'évolution de son état à la Salle Virtuelle
- elle réagit aux actions des Participants dans la Salle Virtuelle

Une instance de la classe Activité constitue un pont vers :

- le modèle d'Information, par les messages, conversations et documents sur lesquels elle agit,
- le modèle de Communication, par sa coopération avec l'objet Médiateur qui lui est associé

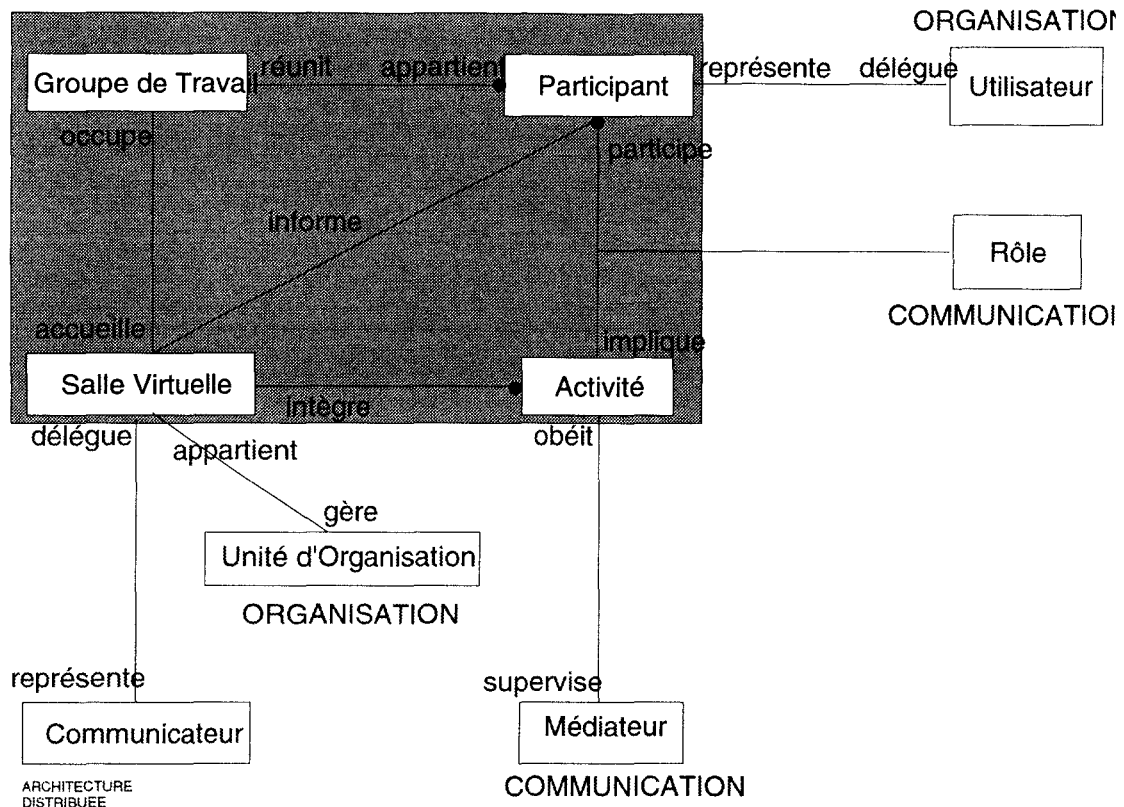


Figure 54. Le modèle de la Salle Virtuelle dans le Système d'Information de Co-Learn

2.2. Le domaine de la Communication

Toutes les activités sont programmées sur le serveur qui gère et mémorise le rôle de chaque participant dans l'utilisation de l'outil, ainsi que l'état et les données associés à l'activité auquel l'outil permet de participer. La réalisation d'une activité est distribuée dans plusieurs types d'objets du système d'information : les activités elles-mêmes mais aussi l'ensemble des rôles que les participants jouent dans les activités et l'ensemble des médiateurs qui supervisent les activités.

Puisque ces classes définissent comment les messages et événements sont produits et consommés par les activités, ces objets font partie du modèle de communication de notre système d'information.

2.2.1. La classe Rôle

Une instance de rôle communique et mémorise les droits dynamiques (types d'action qu'il a le droit d'opérer sur certains types de messages ou de documents) qui changent avec l'état du rôle et la phase de l'activité.

Le rôle offre un pont vers le modèle de la Salle Virtuelle par sa connaissance du Participant auquel il est attribué

2.2.2. La classe Médiateur

Un objet de type médiateur filtre les messages produits par les participants. Il communique à l'activité les actions valides que demandent ces messages. Il produit également des messages à destination des participants pour les informer des changements de leurs droits.

2.2.3. La classe Activité

Du point de vue de la communication, elle a pour rôle de consommer et produire des messages et des événements.

Elle constitue un pont vers :

- le point de vue de la Salle Virtuelle, par sa connaissance de la Salle Virtuelle dans laquelle elle a lieu,
- le point de vue de l'Information, par les messages et documents qu'elle manipule

Un rôle définit l'ensemble des Messages d'Activité qui peuvent être envoyés ou reçus par les utilisateurs jouant ce rôle. Chaque action pertinente d'un participant dans une activité est traduite par un message semi-structuré typé envoyé au médiateur de l'activité. Le Médiateur réagit aux messages en fonction de la phase de l'activité en exécutant des fonctions génériques de médiation ou spécifiques à la sémantique de l'activité.

Nous avons choisi de systématiser l'approche orientée objet. Ainsi, nous considérons que le changement de comportement d'une entité devrait être réalisé par le changement de l'objet qui le représente dans l'espace des objets. Donc, quand la phase de l'activité change, le médiateur devrait changer sa réaction à tous les messages auxquels il peut répondre. Il décide alors de demander à l'activité de nommer un autre médiateur à sa place. Nous définissons une hiérarchie de classes de médiateurs. Au sommet, la classe de médiateur générique définit le comportement commun à toutes les activités, indépendamment de l'activité. Ces méthodes génériques correspondent aux fonctions génériques des activités de coopération de groupe identifiées dans le chapitre 9 de cette thèse. Pour chaque activité, nous définissons une classe de médiateur abstraite. C'est une sous-classe de la classe Médiateur. Elle définit tous les messages qu'un médiateur de l'activité peut recevoir en provenance des participants, c'est à dire son protocole public. Ce protocole peut définir des comportements communs à tous les médiateurs ou des procédures de traitement d'erreur dans le cas de la réception d'un message invalide dans la phase courante (ceci peut se produire en cas de défaillance de la réalisation du modèle de l'Architecture Distribuée). Une activité peut supporter plusieurs modes de médiation (par exemple, un vote peut être conduit à bulletin secret ou à main levée). Pour définir un nouveau mode, un concepteur peut être assisté par le système qui lui présente l'ensemble des messages auxquels il faut pouvoir répondre (ce sont celles définies dans le médiateur générique de l'activité) et les actions qu'il peut déclencher pour répondre à ces messages (ces actions sont définies dans le médiateur générique de la classe et dans la classe de l'activité).

Les activités de groupe en temps différé sont supportées dans le système d'information par les mêmes classes que les activités de coopération en temps réel. Nous avons donc un modèle de communication unifié intégrant les activités en temps différé et en temps réel. Dans une activité de conversation en temps différé, les actions contrôlées par les composants de coopération (Médiateurs et Rôles) et traduits en messages d'activité sont par exemple :

- répondre à un message de conversation,
- lire un message de conversation,
- créer une conversation

Les messages de conversation n'appartiennent pas au modèle de communication : ce sont des objets du Modèle d'Information.

Dans une activité d'édition coopérative de document hypermédia, des exemples de messages d'activité sont :

- lire un type de document,
- créer un type de document,
- supprimer un document,
- créer un lien
- supprimer un lien

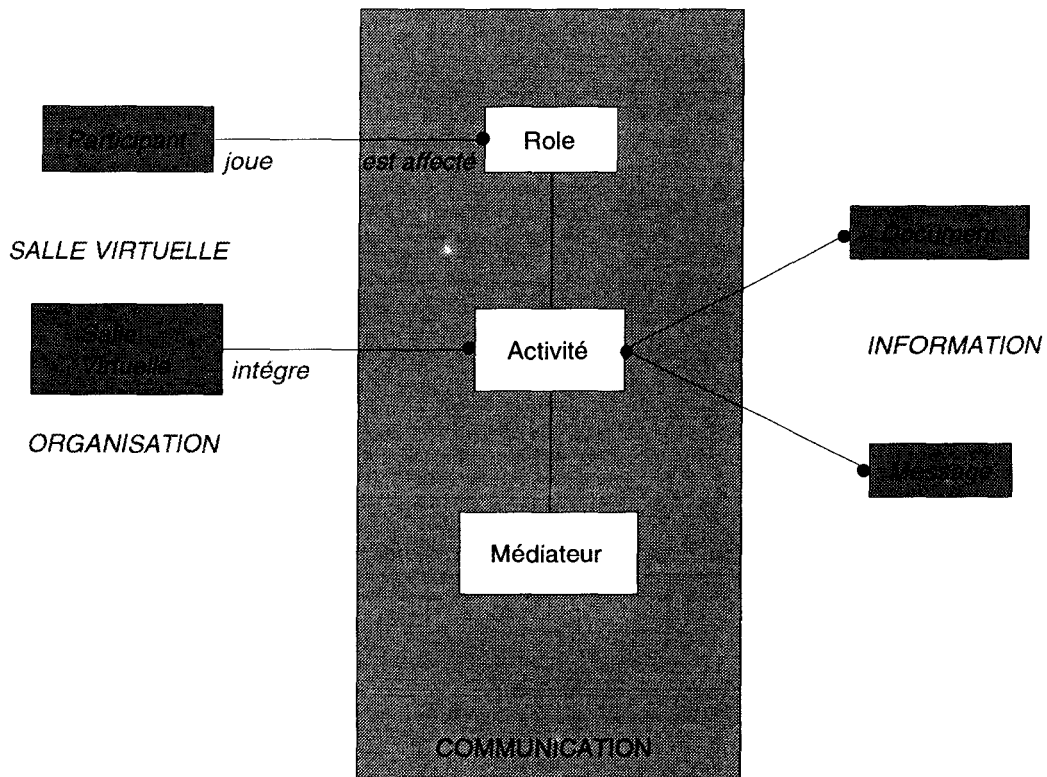


Figure 55. Le domaine de la communication dans le Système d'Information de Co-Learn

2.3. Le domaine de l'Organisation

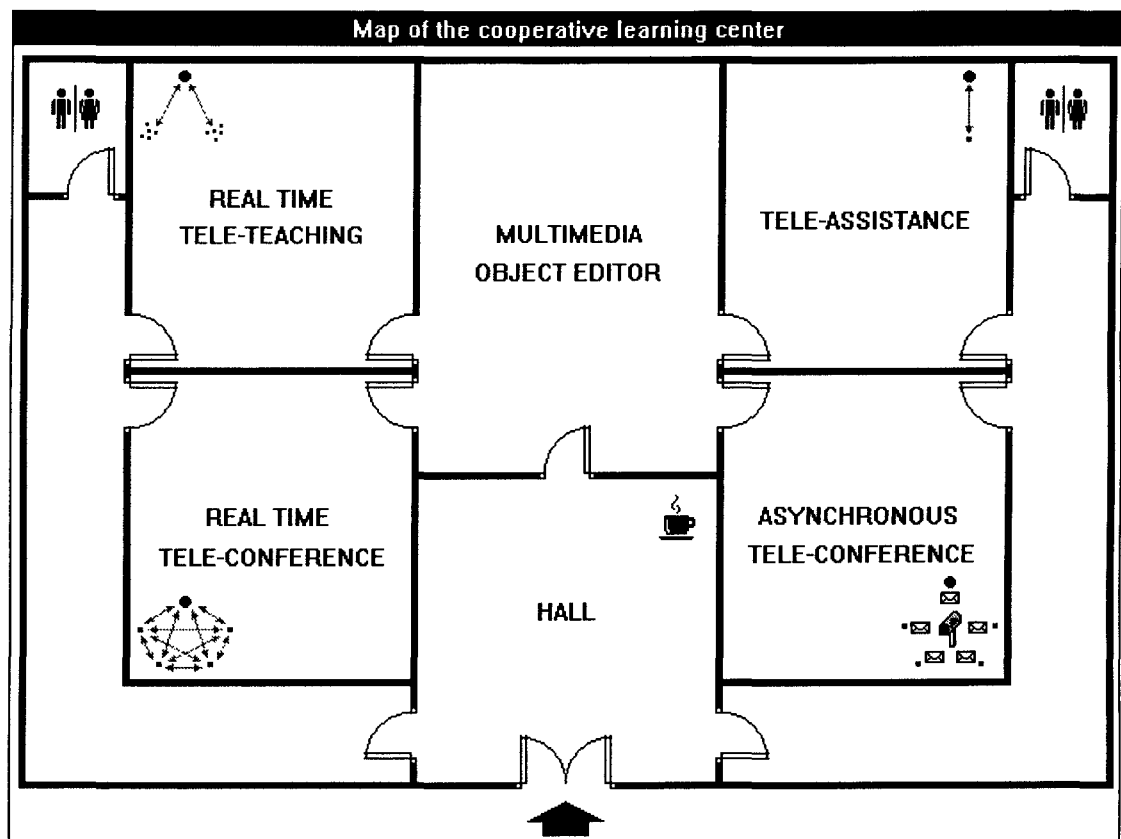


Figure 56. Le plan d'un centre de ressources virtuelles Co-Learn

Comme identifié par MOCCA et [HENNE 94], la base de connaissance organisationnelle contient l'information à propos des objets qui appartiennent à une organisation. Dans une organisation d'apprentissage coopératif, cette base inclue les objets sociaux tels que les enseignants, les apprenants, les rôles organisationnels (par exemple, responsable de département) et les groupes d'apprenants. Elle contient également des ressources telles que les cours, les types de documents, les Salles Virtuelles, les tâches et activités et les coordonnées des utilisateurs (par exemple, le bureau, le téléphone, l'adresse électronique des enseignants et apprenants). La base de connaissance de l'organisation inclue aussi les relations entre ces objets : inscription des participants aux groupes et aux cours, participations dans les Salles Virtuelles et rôles des participants dans les activités. La base d'information distribuée permet aussi le nommage des objets partagés.

Il existe un type spécial de Salle Virtuelle appelé Hall (figure 56) qui propose des outils de navigation dans l'organisation, entre autres un explorateur social utile pour trouver toute l'information sur les autres utilisateurs de l'Organisation : rôles structurels, photo, coordonnées, rôles joués dans les activités et les Salles.

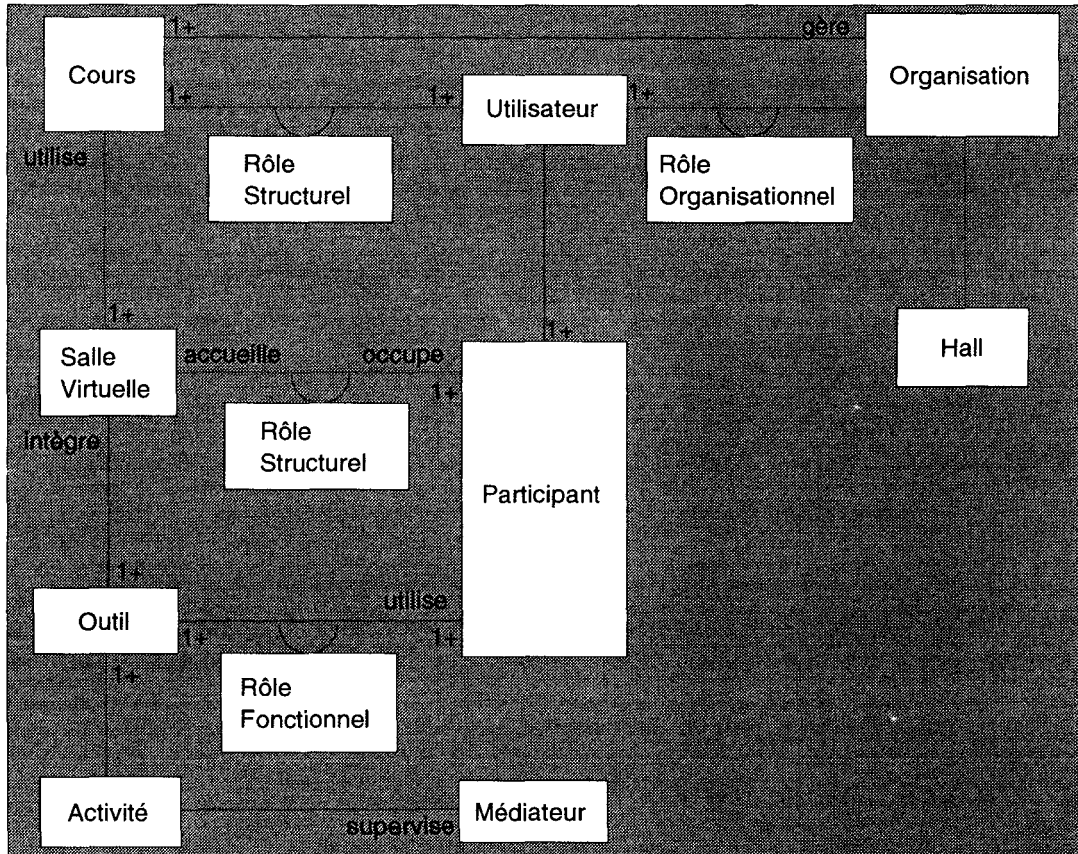


Figure 57. Le schéma générique du domaine de l'organisation du modèle d'information

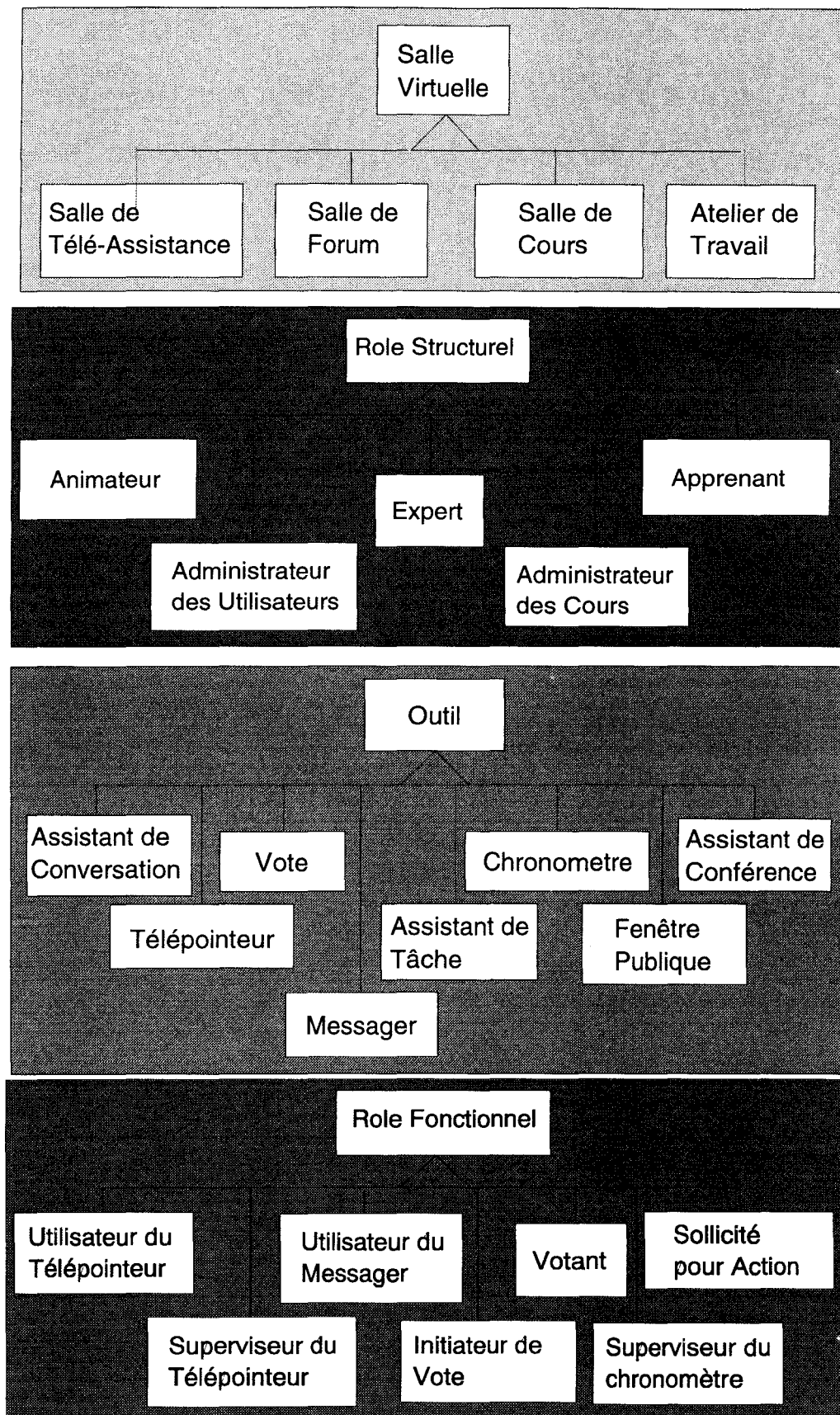


Figure 58. Spécialisation des classes génériques du domaine de l'organisation

2.4. Le domaine de l'Expertise Utilisateur

Le domaine de l'expertise utilisateur contient des connaissances implicites et explicites sur les utilisateurs du système qui peuvent améliorer l'interaction avec le système et le groupe : connaissances, préférences, rôles, tâches et activités en cours, niveau d'expertise. Ce domaine est en étroite relation avec le domaine organisationnel.

La connaissance des utilisateurs est répartie dans le système d'information dans les instances de la classe Utilisateur, Participant, Rôle Structurel et Rôle Fonctionnel. Une instance de la classe utilisateur connaît des informations sociales sur l'utilisateur : nom, prénom et un document hypermédia, constituant une carte de visite évoluée que l'utilisateur peut créer et compléter pour se présenter aux autres membres du groupe. Ceci est semblable aux documents HTML que les utilisateurs de l'Internet prennent l'habitude de publier sur le World Wide Web (réseau hypermédia distribué sur le réseau Internet). Cette information est également comparable à celles qui décrivent les utilisateurs dans le Service de Répertoire X500.

Les rôles structurels joués par un utilisateur reflètent son statut dans l'organisation et induisent des règles de comportement à son intention. Les instances de Participant sont consultées pour connaître l'implication d'un utilisateur dans les Salles Virtuelles de l'organisation : elles renseignent sur son adhésion aux groupes de l'organisation. Les instances de Rôles renseignent sur la participation de l'utilisateur aux activités de l'organisation. Beaucoup d'informations sont contenues dans les relations entre les instances de participant et les instances d'éléments d'information de l'organisation : l'auteur d'un message, les lecteurs d'un message, le responsable d'une tâche, l'ordonnateur d'une action, le responsable d'une action. Ces informations permettent aux personnes de choisir leurs collaborateurs en fonction de leurs compétences et de leur expérience et aident les participants à se coordonner.

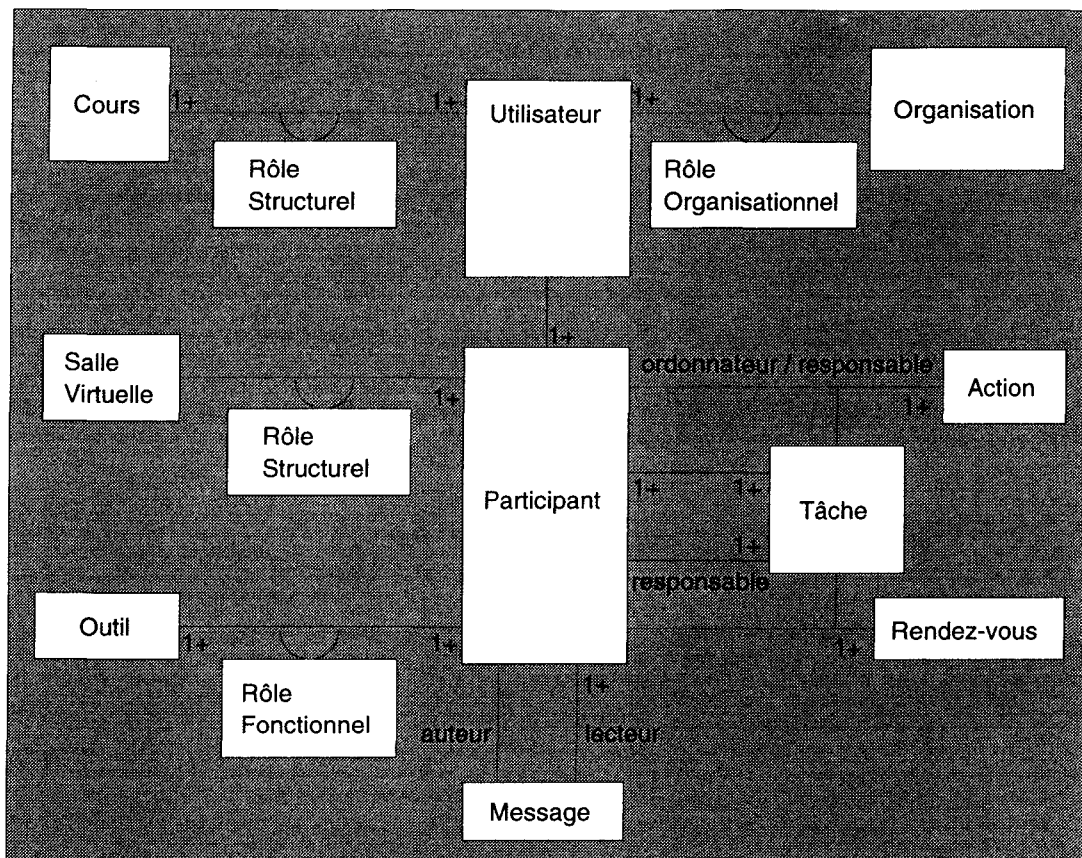


Figure 59. Le domaine de l'expertise utilisateur du modèle d'information

2.5. Le domaine de l'Information

A l'intérieur du domaine d'Information, nous identifions deux espaces : l'espace de conversation et l'espace de documentation, que nous appelons aussi espace d'information (Cf. chapitre 2). L'espace de conversation comporte les messages et les échanges ainsi que les liens divers entre les messages : ce sont les éléments de la conversation. L'espace de documentation contient des objets multimédia et des liens hypermédia entre ces objets : ce sont les éléments de documentation et les objets de la conversation (ce dont on parle).

Les éléments de conversation sont manipulés par les Activités de Conversation. Les éléments de documentation sont manipulés par les Activités d'Édition et de Visualisation.

Les utilisateurs accèdent aux objets d'information par l'intermédiaire des outils. Les outils coopèrent entre eux par l'échange d'objets d'information (figure 60).

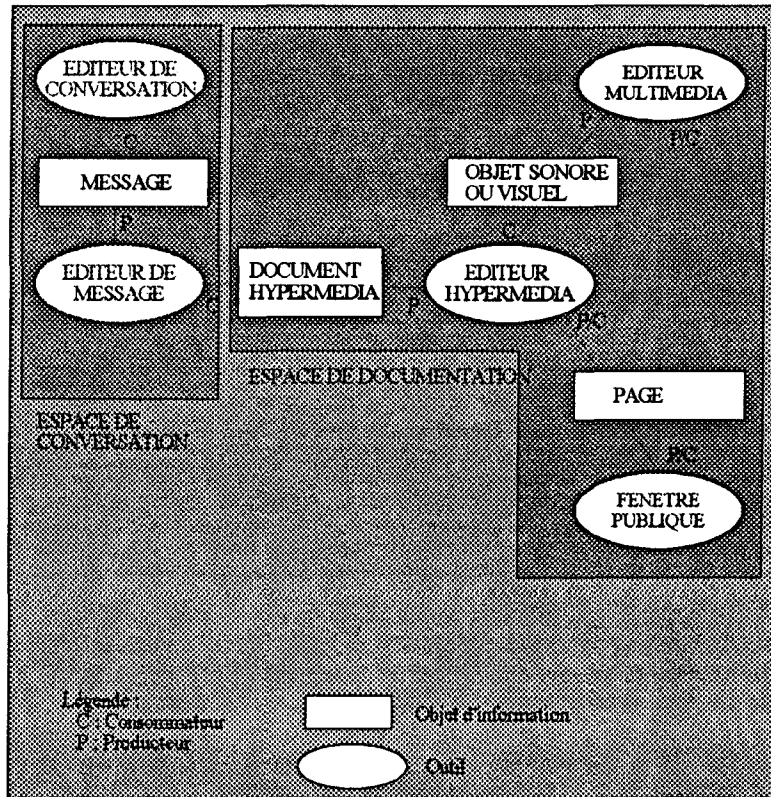


Figure 60. Les relations entre les outils médiatisés par les objets d'information

CHAPITRE 12

LES POINTS DE VUE DYNAMIQUES

Le point de vue dynamique considère la création de réseaux complexes d'objets coopérants et la construction et la configuration de ces objets.

Les objets coopérants dans Co-Learn sont :

- les outils
- les agents de rôle

1. DE L'APPLICATION INTERACTIVE PERSONNELLE AU SYSTEME D'INFORMATION COOPERATIF

1.1. Les applications interactives mono-utilisateur

Une application interactive bien structurée sépare la partie Présentation et le Noyau Fonctionnel. La Présentation a pour rôle de proposer à l'utilisateur une perception de l'application qui est adaptée aux caractéristiques de l'utilisateur et à son environnement. Ainsi, l'aspect de l'interface d'une application pourra varier selon la configuration de l'environnement de travail de l'utilisateur et la configuration de la machine qu'il utilise pour exécuter l'application. Avec les interfaces nouvelles, la Présentation pourra aussi prendre en compte les compétences de l'utilisateur et les caractéristiques de l'environnement, par exemple pour choisir la modalité d'interaction avec l'utilisateur. La partie Contrôleur de Dialogue [COUTA 91] a pour rôle de répercuter sur le Noyau Fonctionnel les actions de l'utilisateur identifiées par la Présentation. Le Contrôleur de Dialogue peut être réalisé par un protocole, comme c'est le cas pour les applications conçues pour l'environnement graphique X, standard de fait sur les stations de travail développé par le MIT. Toute action de l'utilisateur sur la Présentation est analysée, traduite et transmise au noyau fonctionnel par le Contrôleur de Dialogue. Toute modification du noyau fonctionnel est traduite et transmise à la Présentation par le Contrôleur de Dialogue.

En principe, le but de cette décomposition logique est de garantir la réutilisabilité de la Présentation avec d'autres applications ainsi que la réutilisabilité de l'application avec d'autres Présentations. Dans la réalité des systèmes actuels (X sur station de travail, Windows sur PC), la décomposition est exploitée pour réutiliser les composants de Présentation : les environnements graphiques offrent des composants qui prennent en charge les mécanismes d'entrée/sortie. Le développeur d'application interactive est donc déchargé de la tâche de gestion des événements d'entrée sortie de bas niveau : il peut se concentrer sur la logique de l'application et le choix des composants de Présentation les mieux adaptés. Cependant, l'application, elle, n'est pas réutilisable avec une autre Présentation car elle est adaptée aux composants de Présentation qu'elle a choisis : elle comprend et respecte le protocole défini pour ces composants. De plus, les développeurs d'applications interactives constatent que les environnements graphiques actuels ne facilitent leur travail que quand les composants proposés correspondent aux applications développées mais le compliquent beaucoup lorsque ce n'est pas le cas.

Le problème vient de ce que le Contrôleur de Dialogue n'est pas un composant actif qui réalise l'adaptation dynamique de l'application avec la Présentation mais est un protocole que le Noyau Fonctionnel doit respecter. Or ce protocole est indépendant de l'architecture physique et de certains paramètres d'aspect des composants de Présentation mais il dépend des composants de Présentation choisis. En fait, les environnements graphiques standards offrent une indépendance par rapport à la Présentation, mais pas par rapport à la Vue.

Le modèle MVC (Modèle, Vue, Contrôleur) de Smalltalk réalise l'indépendance du Noyau Fonctionnel par rapport à la Vue. Le Noyau Fonctionnel, qui est réalisé par le composant Modèle peut être réutilisé avec des Présentations totalement différentes. Une présentation est réalisée en Smalltalk par un couple Vue-Contrôleur : la Vue affiche une vue du modèle à l'écran, le Contrôleur interprète les actions de l'utilisateur sur la Vue. Lorsque le Modèle est modifié, il avertit tous les objets qui dépendent de lui par le message "update: aSymbol". Les Vues du modèle, qui appartiennent à la liste des dépendants, doivent analyser le paramètre aSymbol pour savoir si elles sont concernées et de quelle manière. Elles vont alors demander au Modèle les renseignements nécessaires à leur mise à jour. Nous constatons que cette fois, ce sont les composants de Présentation (couples Vues-Contrôleurs) qui ont la responsabilité de respecter le protocole imposé par le Modèle. Ainsi, l'environnement Smalltalk offre des classes de composants de présentation standards, dont les instances savent coopérer avec certains modèles standards, mais qui doivent être dérivées et adaptées pour s'interfacer à des modèles dérivés par le développeur de ces Modèles standards.

En conclusion, en retirant au Noyau Fonctionnel la responsabilité de respecter un protocole imposé par la présentation, MVC permet la réutilisabilité du Modèle avec d'autres Présentations. Mais en imposant aux composants de Présentation la responsabilité de respecter un protocole défini par le Noyau Fonctionnel, MVC interdit la réutilisabilité des composants de Présentation. Cependant l'approche orientée objet de Smalltalk permet de dériver facilement des classes de composants de Présentation.

Pour obtenir une réelle indépendance réciproque de la Présentation et du Noyau Fonctionnel, nous pensons qu'il est indispensable que le Contrôleur de Dialogue soit réalisé par un composant actif qui *assure les mécanismes d'indirection et de traduction entre les deux mondes* [COUTA 91]. C'est pourquoi nous pensons qu'un modèle multiagent tel que PAC [Ibid.] est nécessaire à la conception des applications interactives.

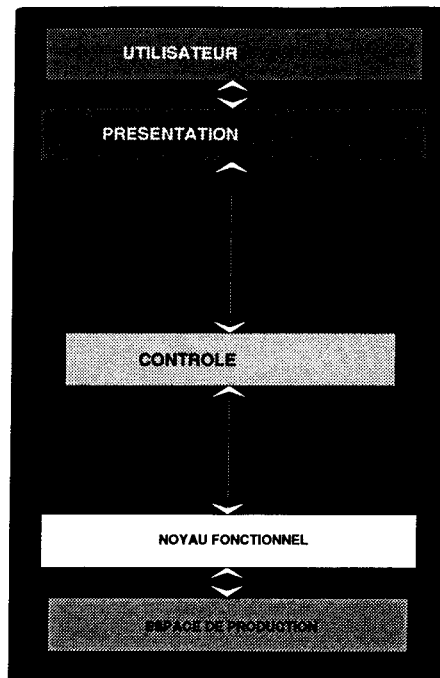


Figure 61. Architecture d'une Application Interactive Mono-Utilisateur

1.2. Les systèmes de partage de vue

Les systèmes de partage de vue permettent à plusieurs utilisateurs d'avoir la même Présentation d'une application, sans devoir modifier l'application. Les utilisateurs se trouvent devant la même présentation. Tous les messages de contrôle envoyés par le Noyau Fonctionnel à destination de la Présentation sont diffusés à toutes les copies de la Présentation. Cependant, les actions des utilisateurs sont filtrées différemment : il est possible d'ignorer les actions d'un utilisateur. Ceci permet de ne permettre qu'à un utilisateur de modifier le Noyau Fonctionnel. En agissant sur ces filtres, les utilisateurs peuvent se passer le contrôle de l'application.

Le mécanisme de partage de vue a pour avantage de permettre de partager très facilement toute application X. Malheureusement, à cause des contraintes expliquées précédemment, les utilisateurs ne peuvent pas avoir des vues différentes de l'application : le principe WYSIWIS ne peut pas être relaxé.

Non seulement tous les participants ont la même vision de l'application mais en plus ils disposent tous des mêmes possibilités d'action (ou aucune). Il est impossible de faire évoluer une Présentation indépendamment des autres. Par exemple, dans le cas d'une application de Vote, il serait intéressant de désactiver la fonction de vote d'un participant dès qu'il a voté (et activer la fonction de consultation des résultats), mais uniquement sur sa présentation locale, et pas sur celles de ses collaborateurs : ceci n'est pas possible avec un mécanisme de partage des Vues.

L'inconvénient est qu'il n'est pas possible de définir de protocoles de coopération et des rôles: soit un utilisateur a tout le contrôle de l'application, soit il n'en a pas du tout et le tour de parole ne peut pas être géré par le système en fonction de l'évolution de l'application : c'est aux utilisateurs de passer explicitement la main. Il n'est pas possible non plus de définir de politique de médiation des conflits.

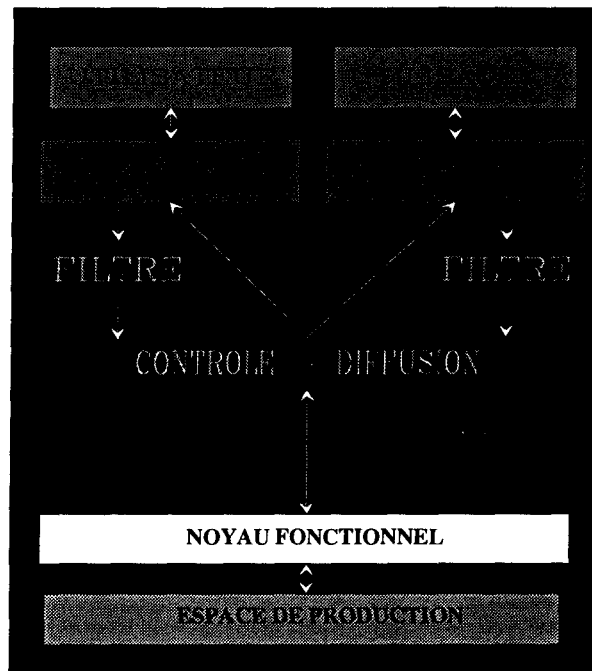


Figure 62. Architecture du Partage de la Vue d'une Application Interactive

1.3. Les applications interactives de groupe

Suite à cette analyse, nous avons conçu une architecture (figure 63) pour les applications interactives de groupe. Nous ajoutons trois composants par rapport à l'architecture de partage de Vues : les Rôles et la Médiation sont des composants actifs, la Coopération est un composant passif.

Un Rôle est un agent qui reçoit la connaissance des règles d'évolution des droits du participant (que nous appelons Règles de Conversation) en fonction de l'évolution de l'Etat Personnel du rôle du participant et de l'Etat Global de l'Activité¹. Cette connaissance lui est transmise à sa création par le composant de Médiation. Il est ensuite averti en permanence par la Médiation de l'évolution de ces états. Il déduit à chaque fois les nouveaux droits (de consultation, d'action) du participant et les répercute sur la Présentation.

¹Nous appelons Activité l'ensemble composé du Noyau Fonctionnel et de la Médiation

Le composant Présentation est modifié dans la mesure où il doit comprendre un deuxième protocole : le protocole de coopération. C'est le protocole qui existe entre l'Agent de Rôle et la Présentation. L'Agent de Rôle envoie des messages indiquant à la Présentation l'ajout ou la perte de droits de l'utilisateur. Si la Présentation n'est pas capable d'interpréter ces changements de droits, il faut intercaler entre la Présentation et l'Agent de Rôle un Traducteur, composant actif qui prend en charge la traduction des ordres de l'Agent de Rôle, qui n'apparaît pas sur notre figure. Il faut noter que la Présentation n'envoie pas de message à l'Agent de Rôle : la relation entre l'Agent de Rôle et la Présentation est de type client-serveur (éventuellement médiatisée par un Traducteur).

Le composant Médiation intercepte tous les messages de toutes les Présentations à destination du Noyau Fonctionnel. Il a la responsabilité d'agréer les Actions du Participant, de résoudre les conflits éventuels. Si l'action est valide dans l'Etat Global actuel de l'Activité, le composant Médiation répercute sur le Noyau Fonctionnel les conséquences de l'Action du Participant. De plus, prenant en compte le Rôle du Participant qui a produit l'Action, l'Etat Personnel de ce Rôle et l'Etat Global de l'Activité, le Médiateur utilise les Règles de Médiation pour modifier l'Etat Global de l'Activité, l'Etat Personnel de certains Rôles et/ou le Rôle de certains Participants. Il avertit les Agents de Rôle concernés de la modification du Rôle ou de l'Etat Personnel du Rôle du Participant qu'ils représentent grâce au composant Coopération.

Le Noyau Fonctionnel n'est pas modifié dans notre architecture : l'avertissement de la Présentation lors de la modification du Noyau Fonctionnel est toujours assuré par le composant Contrôle et est diffusé à toutes les Présentations. Cependant, une Présentation ne fait apparaître à l'utilisateur que la modification des données que l'Agent de Rôle lui a autorisées à montrer. Si la Présentation n'est pas conçue pour filtrer ces modifications, il faut alors que le Traducteur s'interpose également entre le Contrôle et la Présentation pour assurer l'obéissance aux instructions de l'Agent de Rôle.

Le composant de Coopération est réalisé par un Protocole permettant de communiquer à chaque Agent de Rôle les Règles de Conversation du Rôle et les modifications de l'Etat Global de l'Activité ou de l'Etat Personnel du Rôle.

Le composant Coopération est intégré dans un minisystème de communication de groupe, offrant essentiellement des services de distribution. Ce service de distribution permet de désigner des sous-groupes de Participants et particulièrement l'ensemble des participants de l'Activité qui jouent le même Rôle.

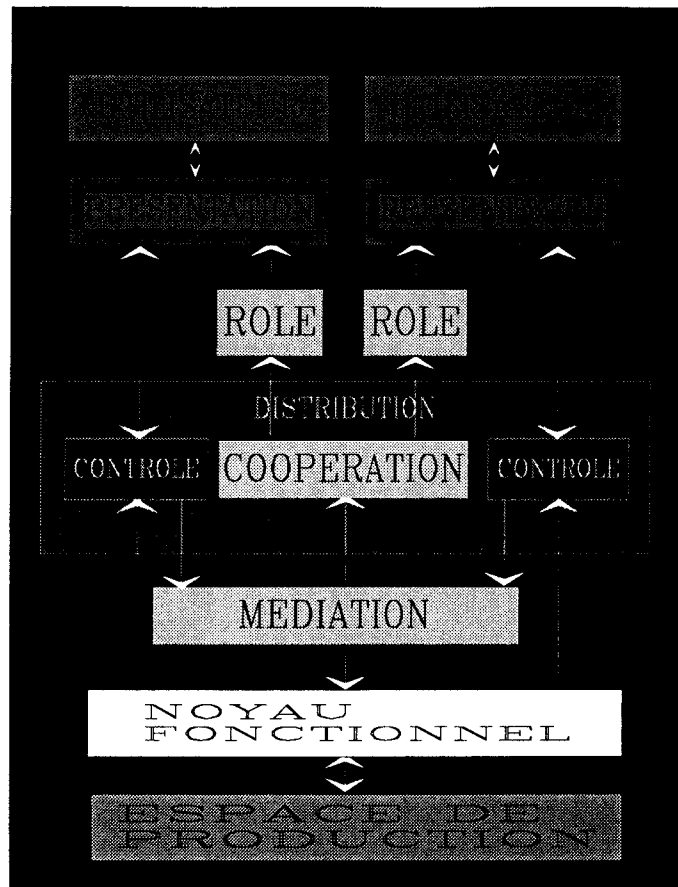


Figure 63. Architecture d'une application interactive de groupe

1.4. Les espaces de coopération

Après avoir défini l'Architecture d'une Activité de Groupe, nous l'intégrons dans un Espace de Coopération.

Un Espace de Coopération (que nous appelons aussi Salle Virtuelle) réunit un ensemble de Participants et un ensemble d'Activités de Groupe.

Le Travail du Groupe est composé de l'ensemble des Activités du Groupe dans l'Espace de Coopération, c'est à dire de l'ensemble des couples (Noyau Fonctionnel, Médiation).

L'ensemble des Rôles d'un Participant (chaque Participant a au plus un Rôle dans chaque Activité) constitue sa Participation au Travail du Groupe.

L'ensemble des Participants constitue le Groupe.

L'ensemble des Participants et leurs Rôles forme le Groupe de Travail.

Tous les Noyaux Fonctionnels accèdent à l'Espace de Production Partagé du Groupe. Cet Espace de Production inclue :

- L'espace d'Information Partagé, qui contient tous les documents et objets informatifs produits ou utilisés par le Groupe ou référencés comme Objets de la Conversation du Groupe,
- L'espace de Conversation du Groupe qui mémorise les Eléments de Conversation du Groupe, ainsi que les liens ente eux et les Echanges qui les structurent,
- L'espace d'Actions Partagé où sont mémorisées les actions promises, demandées ou réalisées par les Participants ainsi que les objets de pilotage des effecteurs éventuels.

Nous avons conçu cette architecture pour supporter à l'intérieur d'un même Espace de Coopération tous les types d'Activités de Coopération : Activités synchrones ou asynchrones, de Conversation, d'Action ou de Coordination.

Sur la figure 63, le rectangle central où s'entrecroisent les canaux de communication, représente le Système de Communication de Groupe. Par souci de clarté, nous n'avons pas représenté sur ce schéma les composants Coopération et Contrôle, mais ils sont intégrés dans le Système de Communication de Groupe. Ce Système de Communication de Groupe doit permettre d'identifier et localiser les différents Participants, les différents Rôles, les différentes Activités et les différentes Présentations pour leur communiquer les messages et communiquer au récepteur l'identité de l'expéditeur d'un message. Aux Médiations, il offre des fonctionnalités de Distribution. Aux Noyaux Fonctionnels, il offre des fonctions de Diffusion.

Nous avons gardé implicites certaines idées représentées sur nos figures jusqu'à présent : les composants Présentation, Rôle, Contrôle, Coopération sont des composants volatiles. En fait, seuls les composants dans la moitié inférieure des figures (Noyau Fonctionnel, Espace de Production, Médiation) sont évidemment persistants : il est évident qu'ils continuent à exister quand un Participant quitte une Application, ou l'Espace de Coopération, du moins tant qu'il existe au moins un Participant dans la Salle Virtuelle en train d'interagir avec les applications.

Dans les applications de partage de Vues, lorsque tous les Participants ont quitté la conférence, plus rien ne persiste des rôles qu'ils ont joués.

Dans notre cas, nous voulons gérer des Espaces de Coopération Persistants. C'est pourquoi nous préférons la métaphore de Salles Virtuelles à celle de conférence: lorsque les Participants quitte une Salle, la Salle et sa configuration peuvent persister. Tous les outils de la salle reste en l'état où le Groupe les laisse.

D'autre part, les entités Participant et Activité, bien que n'apparaissant pas dans l'architecture fonctionnelle sont au coeur du modèle.

Nous gérons donc l'explicitation et la persistance des objets Participant, Rôle et Activité.

Il faut également gérer les allers et venues des Participants dans la Salle Virtuelle. Lorsqu'un Participant entre dans une Salle Virtuelle, il faut configurer sa station en chargeant les Présentations et les Rôles adéquats. Il faut également répercuter les conséquences de cette arrivée sur les Activités de Groupe, et donc sur les Présentations des Participants déjà présents. Pour ceci, un composant essentiel doit être ajouté à l'architecture : la Salle Virtuelle. La Salle Virtuelle existe à travers deux agents :

- L'Agent Privé de Salle Virtuelle représente la Salle Virtuelle auprès d'un Participant. C'est lui qui est chargé d'exécuter les ordres de l'Agent Partagé de Salle Virtuelle : il lance les Agents de Présentation et les Agents de Rôles et il charge les bases de connaissance des Rôles. Il joue ensuite un rôle de distribution des messages en provenance du Système de Communication vers les Agents de Rôle et les Agents de Présentation avec lesquels il maintient des liens de communication.

- L'Agent Partagé de Salle Virtuelle gère une partie de l'Espace d'Organisation qui lui permet de déterminer les Rôles des participants de la Pièce et les Activités qui ont lieu dans la Pièce. Il avertit les Activités concernées des arrivées et des départs des Participants.

Par ailleurs, en l'absence d'un Participant, il faut maintenir l'évolution de ses Rôles dans les Activités en fonction des Actions des Participants présents. Nous avons également vu que le Médiateur a besoin de connaître et de modifier les Rôles des Participants. Pour accéder à ces Rôles, la Médiation s'adresse à l'Agent Partagé de Salle Virtuelle.

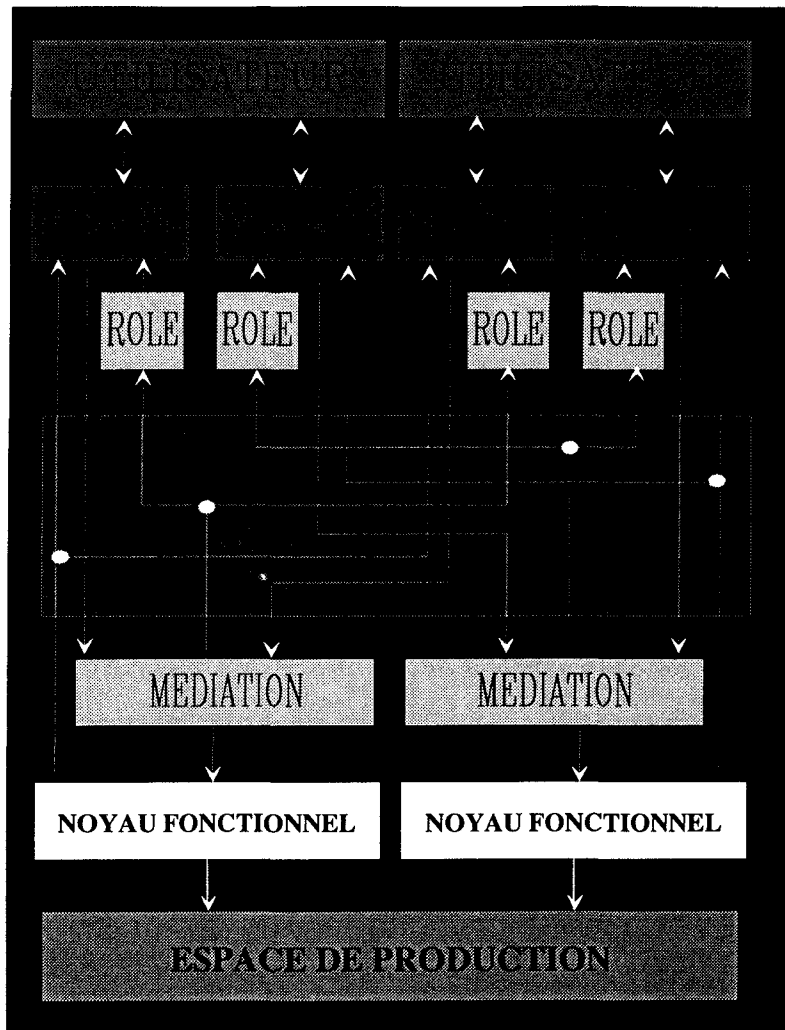


Figure 64. Architecture d'un espace de coopération

2. ARCHITECTURE LOGICIELLE DISTRIBUEE DE L'ENVIRONNEMENT DE COOPERATION CO-LEARN

Au sein d'une Organisation, il existe de multiples Espaces de Coopération. Chaque Espace de Coopération est alors géré par un Agent Partagé de Salle Virtuelle, qui joue le rôle de gestionnaire d'une partie de l'Espace d'Organisation et d'intermédiaire avec la Base d'Objets Organisationnelle. L'Agent Partagé de Salle Virtuelle sait par exemple obtenir auprès de l'Organisation des informations sur les membres de l'organisation, les Salles qui constituent l'organisation, les personnes impliquées dans l'activité d'une Salle avec leurs rôles institutionnels, et leur rôle par rapport aux différents outils qui sont présents dans la Salle.

L'architecture logicielle de Co-Learn est une architecture multi-agent client serveur. Nous avons présenté la répartition des composants d'un Outil Partagé dans cette architecture dans [VIEVI 94]. Pascal Croisy [CROIS 95] a analysé et modélisé la distribution du composant "Atelier de Travail" dans cette architecture, sous l'angle de l'Interface Homme-Machine.

La figure 64 présente cette architecture en se limitant à la représentation des composants nécessaires au support de deux espaces de coopération simultanés, contenant chacun deux utilisateurs présents.

Par rapport à l'architecture présentée par la figure 63, cinq nouveaux composants apparaissent :

Le "Launcher" (L), présent sur chaque site client, accueille le participant dans l'organisation et lui présente de manière intégrée les différents services offerts par le système en lui permettant de naviguer entre les Salles Virtuelles. Le Launcher donne également accès aux services d'administration de l'organisation. Ce composant communique avec l'Agent Local de Hall (AHL).

L'Agent Local de Hall (ALH), présent sur chaque station cliente, est lancé par le Launcher dès qu'un utilisateur demande à entrer dans une session Co-Learn. Il établit alors un canal de communication avec l'Agent Partagé de Hall présent sur un site serveur. C'est entre ces deux entités de Communication que vont transiter les informations nécessaires à l'administration et à la navigation dans l'organisation. Au moment d'entrer dans une session, l'Agent Partagé de Salle Virtuelle est le seul composant du serveur qu'une station cliente Co-Learn est capable de contacter. C'est par la relation client-serveur entre l'Agent Local de Hall et l'Agent Partagé de Hall que la station cliente obtiendra les renseignements nécessaires pour contacter les autres composants serveurs (les Agents Partagés de Salle Virtuelle). L'Agent Partagé de Hall joue donc le rôle d'un Service de Répertoire Global, similaire au "Global Directory Service" de DCE [BEVER 93]. Pour obtenir ces informations utiles, il communique avec la Base de Connaissance Organisationnelle (BCO) et les Agents Partagés de Salle Virtuelle.

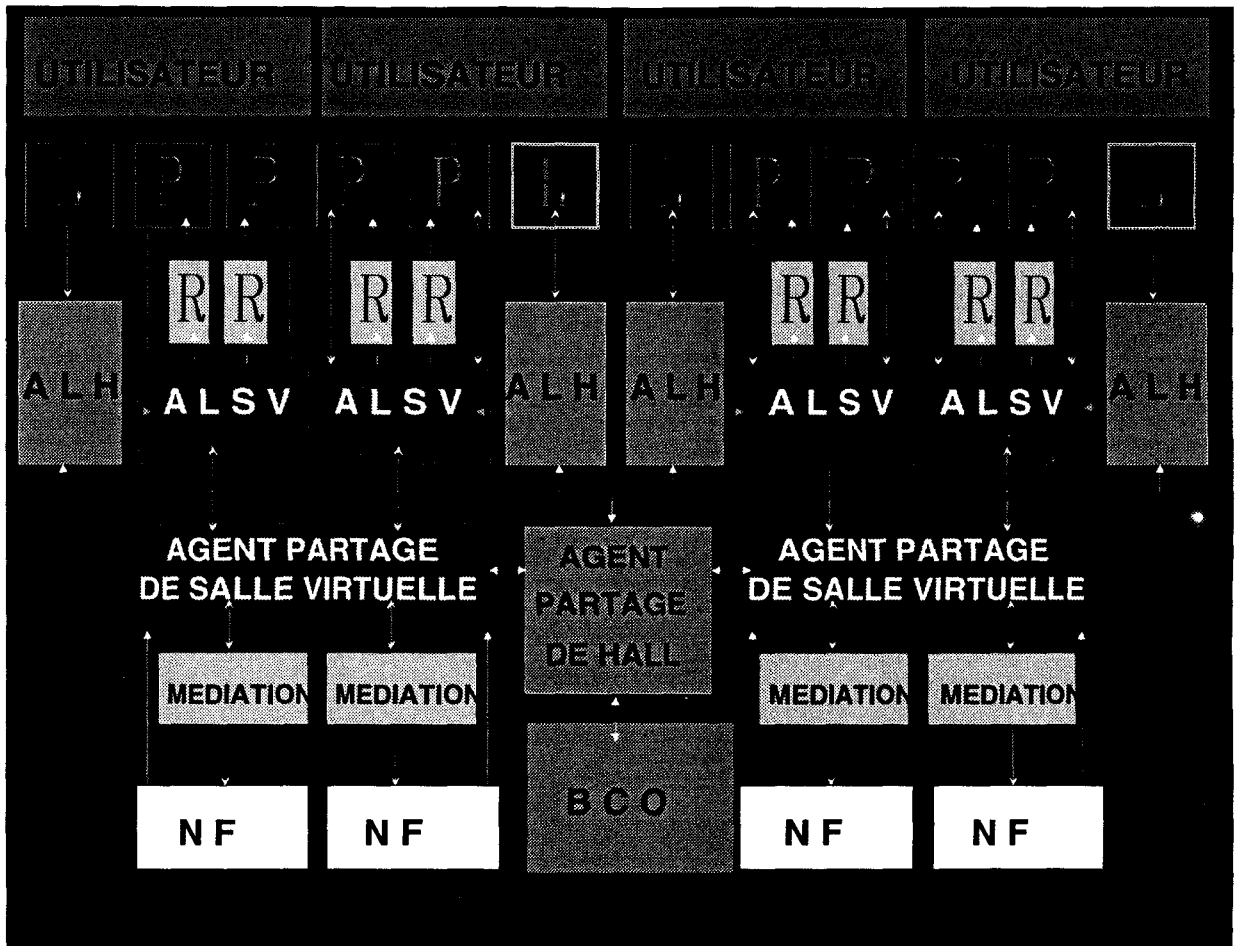


Figure 65. Architecture Logicielle de Co-Learn

Quand l'utilisateur entre dans une Salle Virtuelle, l'Agent Local de Hall lance un Agent Local de Salle Virtuelle auquel il fournit les coordonnées de l'Agent Partagé de Salle Virtuelle qu'il doit contacter. Ce dernier communique à l'Agent Local la liste des outils à offrir à l'utilisateur dans la Salle Virtuelle. Chaque outil lancé par l'Agent Local est composé d'une Présentation (P) et d'un Agent de Rôle (R). L'Agent Partagé de Salle Virtuelle avertit toutes les Activités de la Salle Virtuelle de l'arrivée du Participant. L'Agent de Médiation de chaque activité réagit alors en communiquant le rôle du participant à l'Agent de Rôle. Le Noyau Fonctionnel (NF) de chaque activité transmet à la Présentation les données d'affichage. L'acheminement de ces communications est réalisé par l'intermédiaire des Agents de Salle Virtuelle. La figure 66 montre la réalisation du protocole d'entrée dans une Salle Virtuelle. Le lhm (local hall manager) est l'application Windows réalisant l'Agent Local de Hall. Le lrm (local room manager) réalise l'Agent Local de Salle Virtuelle. L'objet Smalltalk Activity réalise le noyau fonctionnel d'une activité. Les composants Windows P et R sont la réalisation des composants Présentation et Agent de Rôle décrits dans l'architecture logicielle. La figure 66a montre également un objet Smalltalk Role, qui assure la persistance du rôle du participant sur le serveur. Nous faisons également apparaître l'application Windows Updater, mais nous omettons de décrire son implication dans la coopération, afin de préserver la lisibilité du schéma. Dans la réalité, cet outil communique avec son alter ego sur le serveur pour assurer que les outils et les documents installés sur le poste d'accès à partir duquel l'utilisateur s'est connecté sont à jour.

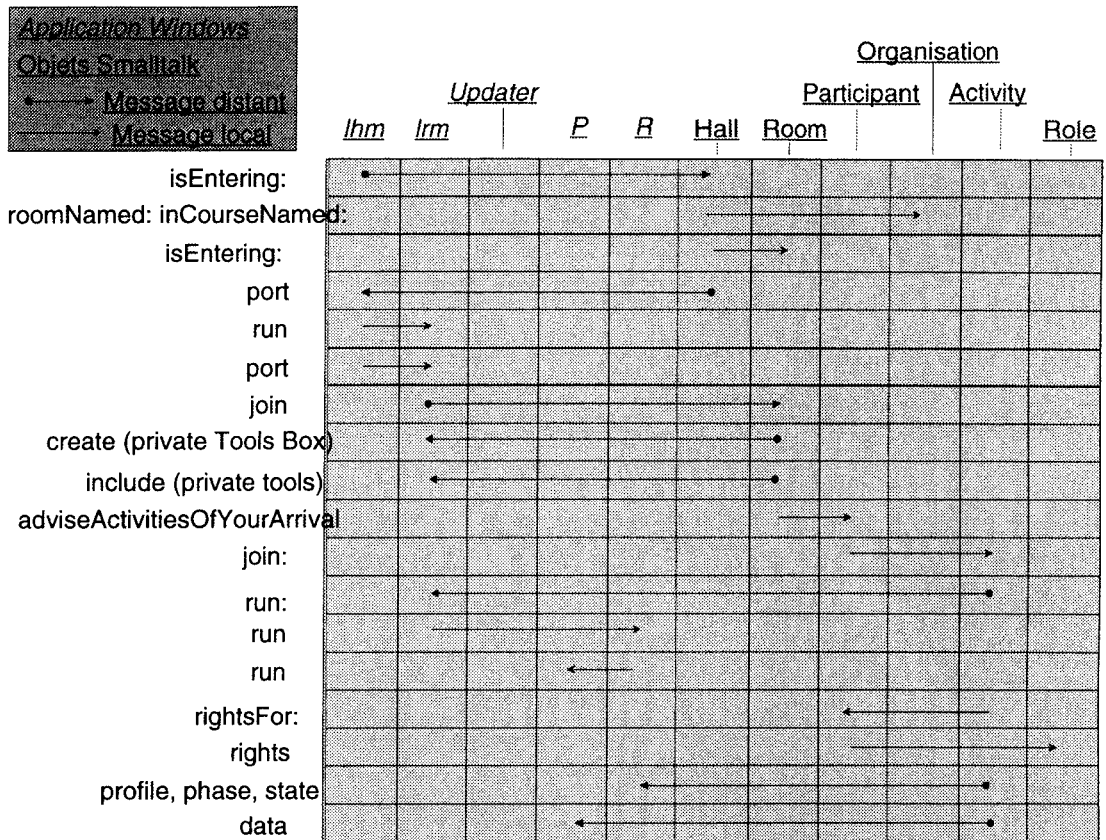


Figure 66a. Coopération des objets distribués Co-Learn lors de l'entrée d'un participant dans une Salle Virtuelle

La figure 66b présente le code Smalltalk qui réalise la coopération décrite par la figure 66a.


```

!Hall methodsFor: 'entering!'

isEntering: aRoomName
  | myRoom myPort |

  myRoom := organisation roomNamed: (aRoomName copyFrom: (1 + (aRoomName
indexOf: $!))
                                to: aRoomName size)
                                inCourseNamed: (aRoomName copyUpTo: $!).
  myRoom isNil iffFalse:
    [DEBUG ifTrue: [TRITextCollector show: requestor user loginName , ' is requesting to
enter in ' , myRoom printString].
    requestor isInAClassRoom ifTrue: [requestor room == myRoom
                                     ifTrue: [self
                                             sendError: 'the user is already registered in this
classroom !!!! We assume he left it improperly'
                                             to: requestor
                                             for: self]
                                     ifFalse:
                                       [self
                                        sendError: 'the user is already registered in another
classroom !!!! We assume he left it improperly'
                                        to: requestor
                                        for: self.
                                        requestor exitClassRoom.
                                       ]].
    requestor room: myRoom.
    myPort := myRoom isEntering: requestor user.
    self
      send: (SimpleDDEMessage withOpCode: 'PORT' with: myPort printString)
asString
  to: requestor
  for: self!]

-----

!Organisation methodsFor: 'queries!'

roomNamed: aRoomName inCourseNamed: aCourseName
  | chosenCourse |
  chosenCourse := CLCourses at: aCourseName.
  chosenCourse isNil iffFalse:
    [^chosenCourse roomWithId: aRoomName ]
  ifTrue: [^nil]!

-----

!CLCourse methodsFor: 'accessing!'

roomWithId: aString
  "Returns the room with the id 'aString'."

  ^roomTypes detect: [ :each | each name asLowercase = aString asLowercase]
ifNone: [nil]!

-----

!ClassRoom methodsFor: 'workGroup Management!'

isEntering: aParticipant
  communicator isNil ifTrue: [self open].
  ^communicator port!

```

```

join
    requestor isConnected ifTrue: ["Le participant est deja connecte dans cette salle
(anormal)"]
        self
            sendError: 'you were already in the room : ', self name
            to: requestor
            for: self].
    requestor connect.
    self sendTeleloadConfigTo: requestor.
    self sendPrivateToolsTo: requestor.
    requestor adviseUpdaterOfYourArrival.
    DEBUG ifTrue: [TRITextCollector show: requestor name , ' vient d"entrer dans la salle
', name; cr]!

```

!ClassRoom methodsFor: 'communication'!

sendPrivateToolsTo: aParticipant

```

| aMessage |
aMessage := SimpleDDEMessage withOpCode: 'user' with: aParticipant name.
self send: aMessage to: aParticipant for: self.
aMessage := SimpleDDEMessage withOpCode: 'create' with: self name.
self send: aMessage to: aParticipant for: self.
aMessage := SimpleDDEMessage withOpCode: 'include' withArgs: privateTools.
self send: aMessage to: aParticipant for: self.!

```

!Participant methodsFor: 'activities Management'!

adviseActivitiesOfYourArrival

```

self activities do: [:each | (each isKindOf: Exchange)
    ifFalse: [(each isKindOf: Updater)
        ifFalse:
            [each join: self]]]!

```

!Activity methodsFor: 'Participants'!

join: t1

```

self runFor: t1.
self sendRoomNameTo: t1.
self sendAllDataTo: t1.
self sendProfileTo: t1.
^self! !

```

!Activity methodsFor: 'Communication'!

sendProfileTo: t1

```

myRoom sendProfileTo: t1 for: self.

```

!ClassRoom methodsFor: 'communication'!

sendProfileTo: aParticipant for: anActivity

```

| myMessage |
(anActivity isKindOf: TaskBrowser)
    ifTrue: [self sendTaskProfileTo: aParticipant for: anActivity]
    ifFalse:
        [myMessage := SimpleDDEMessage profile: (aParticipant rightsFor:

```

```

anActivity).
    self
        send: myMessage
        to: aParticipant
        for: anActivity.
    myMessage := SimpleDDEMessage phase: anActivity phase state:
(aParticipant roleFor: anActivity) state.
    self
        send: myMessage
        to: aParticipant
        for: anActivity]!

!Participant methodsFor: 'queries'!

rightsFor: anActivity
^(roles detect: [ :each | each activity == anActivity ]) rights!

!Role methodsFor: 'accessing'!

rights
^self class rights!

```

Figure 66b. Les méthodes des objets Smalltalk activées lors de l'entrée d'un participant dans une Salle Virtuelle

La figure 67 présente un exemple de coopération entre les objets Co-Learn pour répondre à une requête de l'utilisateur dans le cadre de l'activité de conférence asynchrone. L'utilisateur interagit avec une application Windows, appelée TskBws, qui permet de visualiser et gérer les tâches d'une conférence. Lorsque l'utilisateur demande la description d'une tâche, l'application Windows soumet la requête à l'objet TaskBrowser sur le serveur.

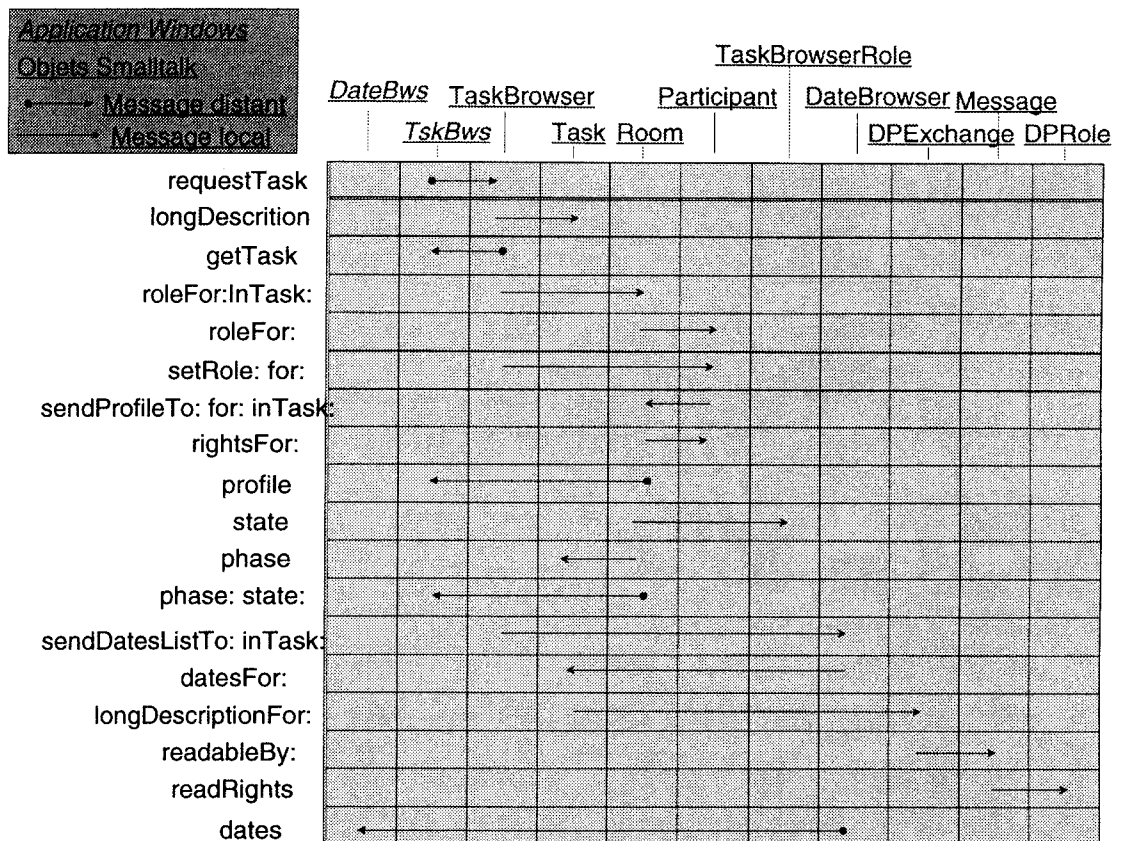


Figure 67. Coopération des objets distribués du réseau Co-Learn lors d'une requête dans l'activité de conférence

L'architecture logicielle de Co-Learn simule un système d'objets distribués. Les messages envoyés par les applications Windows qui réalisent les Agents de Rôles et les Présentations aux objets du serveur (Activités, Salles Virtuelles) produits sous forme de messages DDE sont traduits en messages objets dès leur réception sur le serveur. Réciproquement, les messages envoyés par les objets du serveur aux Agents de Rôle et aux Présentations sont traduits en messages DDE au moment de leur expédition. Cette stratégie nous permet d'envisager un portage des clients sous Smalltalk qui permettrait d'enrichir la conversation entre les clients et les serveurs. En particulier, l'utilisation de Smalltalk sur les clients nous permettrait de configurer finement les clients en créant et en instanciant les classes adéquates, sans rupture sémantique ou syntaxique. Smalltalk ayant été retenu comme langage d'interface, nous pouvons aussi envisager l'utilisation de CORBA pour faire communiquer les serveurs et clients Co-Learn.

CHAPITRE 13**LES POINTS DE VUE DE L'INGENIERIE****1. PROGRAMMATION DES COMPOSANTS DU SYSTEME DISTRIBUE
CO-LEARN****1.1. Programmation des composants logiciels des clients**

Les Présentations, les Agents de Rôle, l'Agent Local de Hall, l'Agent Local de Salle Virtuelle [COLEARN 92c][COLEARN 92f] et les services de communication [COLEARN 93a] sont programmés en C++. Le choix de l'environnement de développement [COLEARN 92a] a été laissé à la responsabilité de chaque partenaire du consortium. Ainsi, certains modules ont été réalisés à l'aide de l'environnement de développement de Borland et la librairie Object Windows, d'autres à l'aide de l'environnement de Microsoft. Quoi qu'il en soit, tous les composants ont été programmés sous forme de Classes. Une description plus détaillée de la réalisation de certains composants des clients (Présentations, Agents de Rôle) est proposée dans [CROIS 95].

Le Launcher [COLEARN 93b] a été réalisé par un partenaire espagnol, Software de Base, en Visual Basic.

1.2. Programmation des composants logiciels des serveurs

Les services de communication ont été écrits en langage C, pour exploiter la gestion de processus du système d'exploitation.

Tous les autres composants du serveur sont réalisés en Smalltalk [GOLDB 83][DUGER 91], à l'aide de l'environnement ObjectWorks Smalltalk-80 de Park Place. Nous avons choisi Smalltalk pour bénéficier des avantages d'un environnement de développement totalement objet, semi-interprété et facilement distribuable. Nous tirons particulièrement profit de l'héritage et du "*late binding*" pour la gestion des rôles et la médiation des activités de coopération. L'aspect interprété du langage et la gestion automatique de la mémoire facilite l'instanciation dynamique de classes (Participants, Rôles, Salles Virtuelles, Activités) ainsi que l'écriture et l'interprétation de scripts. Smalltalk est très adapté aux projets de recherche, développement et évaluation tel que Co-Learn parce qu'il permet une approche incrémentale et modulaire. La modularité et l'évolutivité offertes par Smalltalk sont indispensables pour concevoir coopérativement un système serveur qui doit communiquer avec un ensemble de modules clients développés par des partenaires distants, comme c'est le cas dans Co-Learn. De plus, le serveur Co-Learn lui-même est composé de plusieurs modules développés par Trigone et la Technische Hochschule Darmstadt. Smalltalk nous permet surtout de modifier le système au fur et à mesure de l'expression des besoins de l'utilisateur ou des évaluateurs, dans un domaine où ces besoins ne peuvent être exprimés à l'avance mais où le système donne naissance à de nouveaux besoins, qu'il doit satisfaire. Les fonctionnalités du serveur peuvent devoir être modifiées pour tester des hypothèses des chercheurs en informatique, en psychologie, en sociologie ou en pédagogie. Enfin, la sélection de Smalltalk comme langage d'interface de CORBA permet d'envisager la distribution du serveur dans un avenir proche.

2. TECHNIQUES DE CONTROLE LOGICIEL

2.1. Contrôle logiciel sur les clients

Les applications clientes sont réalisées dans l'environnement Windows [HELLER 92]: elles utilisent donc entre elles un système de contrôle et de synchronisation événementielle.

2.2. Contrôle logiciel sur les serveurs

L'utilisation de processus Smalltalk sur le serveur induit des techniques de contrôle logiciel concurrent. Les objets sont synchronisés à l'aide de sémaphores. Le flux d'exécution suit les envois de messages entre objets. Le système d'objets réalisé n'est pas un système d'objets actifs parallèles car la gestion du partage du temps entre les processus est assez rudimentaire dans l'environnement Smalltalk de base : le processus actif monopolise le temps de traitement tant qu'il ne le libère pas explicitement. Or la programmation par envoi de messages rend difficile le suivi du flux de traitement qui saute d'un objet à l'autre. Nous avons donc adopté une méthode privilégiant la sécurité, peut-être aux dépens de la vitesse de traitement. Le flux de traitement dans le serveur est dirigé par l'arrivée des messages dans les Salles Virtuelles. Les processus de scrutation des salles virtuelles ont tous la même priorité. Quand l'un d'eux obtient la main, il regarde s'il y a un message à traiter dans la Salle Virtuelle. S'il n'y en a pas, il libère immédiatement le processeur. S'il y a un message à traiter, il déclenche son traitement par un envoi de message à la Salle Virtuelle, qui va déclencher une cascade d'envois de message. Lorsque ce traitement est terminé, il libère le processeur.

Les différents processus de la couche de communication s'exécutent de manière concurrente, exploitant la propriété multitâche du système d'exploitation. Ces processus sont consultés à tour de rôle, sur demande du serveur. Chaque processus prend en charge la communication à l'intérieur d'une Salle Virtuelle.

3. TECHNIQUES DE COMMUNICATION

3.1. Techniques de Communication inter-applications sur les clients

Les composants logiciels sur un client utilisent tous les mécanismes de communication offerts par l'environnement Windows. Les mécanismes DDE et OLE [HELLER 92] réalisent des communications par événements et mémoire partagée. L'appel de DLL [PETZO 92] réalise une coopération orientée opérations.

Les liens DDE (échange dynamique de données) sont utilisés entre le Launcher et les Agents Locaux de Hall et de Salle Virtuelle.

La communication entre la Présentation et l'Agent de Rôle d'un Outil est également établie par une conversation DDE.

Les Agents de Rôle et l'Agent Local de Salle Virtuelle coopèrent par **envoi de messages** car ils appartiennent au même espace mémoire.

La couche de communication est réalisée sous forme d'une **DLL** (bibliothèque dynamique). Les applications s'interfaçent donc avec les services de communication par un **appel de fonction**.

La Fenêtre Publique communique avec les éditeurs standards de Windows (Write, PaintBrush, Designer...) par l'intermédiaire du **Presse Papier**.

L'éditeur de Documents Hypermédia [COLEARN 94b] [BORST 94] utilise l'**OLE** pour intégrer les objets multimédia produits par les éditeurs Windows.

Enfin, toute application sous Windows est associée à une fenêtre, qui est une sorte d'objet actif qui reçoit des messages qui sont mémorisés dans la file d'attente des événements de la fenêtre. L'application doit définir le traitement des messages : il s'agit donc d'une **programmation événementielle**.

[CROIS 95, pp. 105-111] présente la réalisation des composants Présentations, en particulier la définition de la classe générique TWindowCoop. Cette classe offre aux instances des classes de fenêtre qui en héritent les fonctionnalités de communication DDE adaptées à l'environnement de coopération.

Goldwing [COLEARN 93c], le composant qui donne accès aux données organisationnelles et documentaires, a été réalisé par la Technische Hochschule Darmstadt (THD) sous la forme d'une **DLL**. Les fonctions d'accès à la base organisationnelle offertes par cette bibliothèque sont utilisées par le Launcher. Le Gérant de Médiathèque utilise les fonctions d'accès à la base de documents.

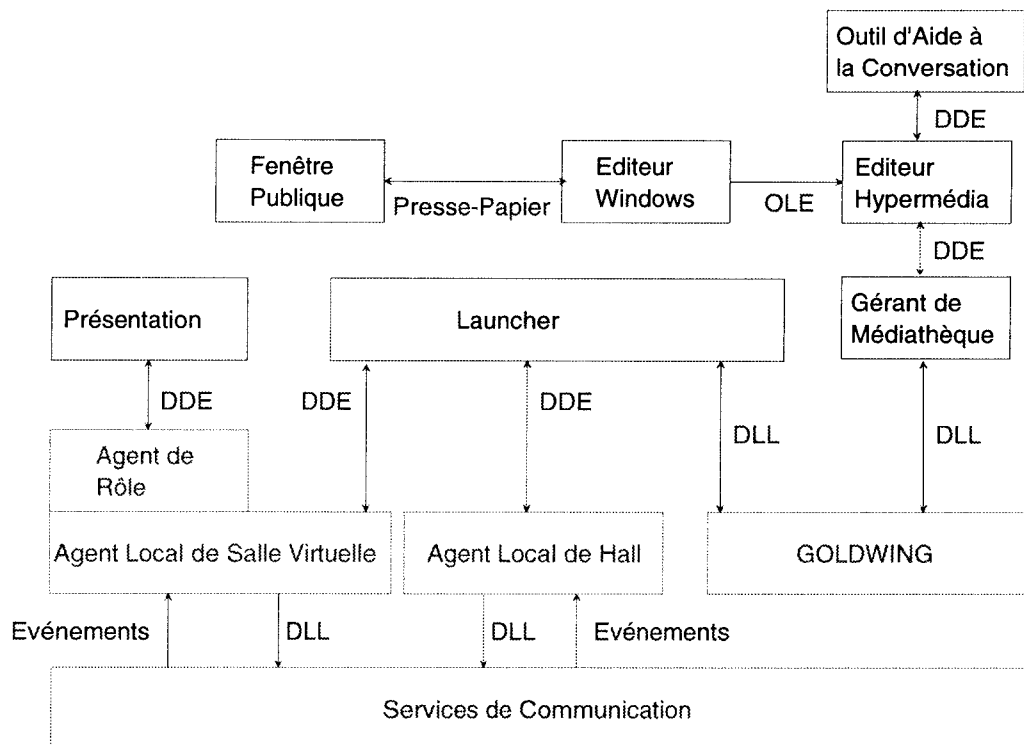


Figure 68. L'architecture de communication sur les clients

3.2. Techniques de communication sur les serveurs

La majeure partie du serveur est intégrée dans une image Smalltalk. A l'intérieur de cette image, les objets, réalisés par le laboratoire TRIGONE et THD, communiquent par l'**envoi de messages objets**.

Les services de communication, réalisés par l'Université Polytechnique de Madrid (UPM) sur nos spécifications [COLEARN 93a], sont constitués d'un ensemble de processus qui communiquent avec la Machine Virtuelle Smalltalk par **sockets** [STEVE 90][RIFFL 95], et d'une librairie de fonctions C liée à la Machine Virtuelle Smalltalk.

Les objets de l'image Smalltalk communiquent avec les services de communication par l'**appel de fonctions** de la librairie CHAPI (Communication Handler Application Programming Interface) [COLEARN 93a]. A cause de problèmes de conflits d'interruption avec la machine virtuelle Smalltalk, nos partenaires d'UPM n'ont pas pu réaliser l'interface événementielle que nous leur avons demandée. Ceci nous oblige à utiliser la technique du polling pour détecter l'arrivée de messages dans une Salle Virtuelle. Ceci qui pourrait altérer sensiblement les performances des serveurs dans le cas d'une utilisation intense (beaucoup de Salles Virtuelles ouvertes à scruter). Une solution partielle et provisoire consiste à interrompre les services de communication des Salles Virtuelles inoccupées, les réactivant lorsqu'un participant demande à y entrer. La meilleure solution à long terme consiste à remonter les services de communication au niveau de l'image Smalltalk : l'utilisation des objets Sockets de Smalltalk permet d'éviter le polling. Nous avons déjà pu vérifier la faisabilité de cette solution au début du projet quand nous avons développé une couche de communication provisoire, dans l'attente de la disponibilité des services de communication développés par nos partenaires.

3.3. Techniques de communication entre les clients et les serveurs

La communication entre les Agents Locaux et Partagés de Salle Virtuelle est actuellement réalisée par circuit virtuel (car ce sont les temps d'établissement des communications qui sont les plus longs sur le réseau RNIS [IRIS 89][DICEN 90]). Les processus de Communication de Salle Virtuelle installés sur les serveurs assurent la sérialisation et la diffusion des messages :

- d'un serveur vers plusieurs clients,
- d'un serveur vers un client,
- d'un client vers un serveur,
- d'un client vers tous les autres clients
- d'un client vers tous les clients et un serveur.

Certains de ces processus sont spécialisés pour le transfert et la diffusion de fichiers.

Certains Outils sont capables d'utiliser un disque réseau partagé via NFS pour échanger et diffuser des documents. Ce sont par exemple la Fenêtre Publique, l'Outil d'aide à la Conversation et les Agents Locaux de Salles Virtuelles qui savent travailler soit avec un disque partagé, soit avec les services de transfert de la couche de communication de groupe de Co-Learn.

4. TECHNIQUES DE GESTION DE LA PERSISTANCE

La persistance est assurée par le couplage de Smalltalk à la base d'objets GemStone¹ de Servio Logic Corporation [MAIER 90][BUTTER 91][PURDY 91]. Le contenu des documents hypermédia n'est pas stocké dans la base mais dans le système de fichiers. Comme la base ne mémorise que la référence des documents hypermédiés et des informations sur les droits d'accès qui leur sont associés en terme de rôle et d'utilisateur Co-Learn, le système est ouvert vers tout système de stockage de documents, par exemple le World Wide Web. Les fichiers exécutables Windows à charger sur les clients lors de leur installation ou pendant leur participation à une session Co-Learn pour la mise à jour de nouvelles versions ou de nouveaux outils sont stockés dans le système de fichiers des serveurs. L'outil de mise à jour (Updater) est capable de les installer sur les postes d'accès en utilisant une copie d'un disque réseau ou un service de téléchargement.

La persistance des objets Smalltalk est assurée en utilisant le mécanisme de **transparence** du GSI (Gemstone Smalltalk Interface). La figure 68 montre comment, dans le script d'installation, une nouvelle organisation créée sur un serveur Co-Learn est rendue persistante. L'objet TheOrganisation, instance de la classe Organisation est la racine du système d'objets. Tous les autres objets sont rendus persistants par propagation de la persistance sur les liens d'agrégation. Par la suite, tous les messages sont envoyés aux objets Smalltalk et les modifications sont répercutées automatiquement sur la copie des objets dans le cache de la base de données, de façon optimale. L'application peut fonctionner en ignorant l'existence de la base d'objets. Il est de plus possible d'envoyer des messages aux objets, activant des méthodes spécifiques à la copie persistante ou commune à l'objet "applicatif" et à sa copie persistante. En effet, il est possible de définir des méthodes différentes dans une classe et dans sa copie persistante. Ceci est utile pour utiliser des méthodes optimisées de recherche d'objets offertes par les classes fondatrices du schéma de la base GemStone ou pour définir des méthodes qui assurent l'intégrité de la base.

¹Cf Annexe 1

```

"Create the new organisation"
Smalltalk at: #TheOrganisation put: (Organisation new: (DialogView request: 'What is the
name of the organisation you want to create ?') asSymbol).

"Make the organisation transparently persistent, by connecting it to the GemStone Database"
Smalltalk at: #TheOrganisationConnector
      put: (GSNameConnector stName: #TheOrganisation gsName: #TheOrganisation).
GSI currentSession addConnector: (TheOrganisationConnector updateGSONConnect;
connect).
TheOrganisationConnector updateSTONConnect.

```

Figure 69. Extrait du script Smalltalk d'installation d'une organisation Co-Learn

Nous n'utilisons le mécanisme transactionnel que pour assurer les principes d'Atomicité, de Cohérence et de Durabilité. Ainsi nous n'utilisons pas le principe d'isolement : nous utilisons la même transaction pour tous les utilisateurs. La coopération des utilisateurs est gérée au niveau des applications, c'est à dire au niveau des objets. La gestion du flux de traitement que nous avons choisie assure la sérialisation des requêtes. Les principes d'Atomicité, de Cohérence et d'Atomicité sont garantis au niveau d'une transaction globale organisationnelle. Une session organisationnelle commence quand le premier utilisateur entre dans l'organisation (dans le Hall). Elle se termine lorsque le dernier utilisateur en sort : l'objet organisation déclenche alors l'arrêt de tous les processus de communication, la sauvegarde de tous les objets de l'organisation dans la base d'objets puis la réinitialisation des processus de communication (figure 70).

```

!Hall methodsFor: 'workGroup Management'!

exit: aParticipant

    aParticipant disconnect.
    DEBUG ifTrue: [TRITextCollector show: (aParticipant user loginName, ' is exiting the
hall'); cr].
    workGroup remove: requestor.
    workGroup markDirty.
    organisation tryToSave.!

!Organisation methodsFor: 'persistence'!

tryToSave
    (self nooneConnected)
        ifTrue:
            [self terminateRoomsCommunication.
             GSI currentSession commit.
             self resetRoomsCommunication.]

```

Figure 70. Extraits du code Smalltalk assurant la persistance sur le serveur

Cette stratégie permet, en vertu du principe d'Atomicité, de garantir que le système d'objets enregistré est toujours cohérent. De plus, notre stratégie nous assure de ne pas perdre une action d'un utilisateur. En effet, lorsque le hall détecte la demande de sortie du dernier utilisateur, il est forcément le seul processus actif et tous les autres processus ont donc achevé leur tâche. Il est vrai que les processus UNIX de communication perdent les messages des utilisateurs qu'ils détiennent dans la file d'attente au moment où l'organisation demande leur réinitialisation mais ces messages ne peuvent être que des demandes d'entrée en session. Au pire, selon le délai d'attente réglée sur le client, le logiciel client signalera à l'utilisateur que la connexion au serveur a échoué et l'utilisateur devra réitérer la demande de connexion.

Par rapport à l'étude que nous avons présentée dans le chapitre cinq et en particulier les deux possibilités que nous avons exposées pour supporter une plateforme de coopération, nous avons retenu la première solution et nous n'avons pas utilisé d'objets actifs, pour les raisons que nous avons également exposées dans le chapitre cinq, à savoir la complexité de gestion de la persistance d'un système d'objets actifs.

Cette stratégie de gestion de la persistance présente cependant un défaut important : un utilisateur ne sait jamais si son travail est enregistré durablement, et ne peut pas être certain qu'il le sera un jour. En effet, la durabilité n'est garantie qu'au niveau d'une session organisationnelle, dont les limites (début et fin) ne sont pas connues des utilisateurs. Il est possible que les objets ne soient jamais enregistrés, s'il y a en permanence une personne connectée au serveur, soit parce qu'une personne ne se déconnecte pas, soit parce que les connexions des différents utilisateurs se relayent. Pour le projet Co-Learn, nous avons adopté une solution simple, qui consiste à obliger les utilisateurs à se déconnecter tous les soirs pour être certain d'enregistrer chaque nuit un état cohérent du système d'information. En conservant la trace des connexions de la journée et des enregistrements, en cas de problème, il est possible d'avertir dès le lendemain les utilisateurs dont les transactions du jour n'ont pas pu être enregistrées. Cette méthode est plus délicate si les fuseaux horaires des utilisateurs ne sont pas compatibles. Or, un utilisateur peut utiliser un ordinateur pour se connecter à Co-Learn à partir de n'importe quel point du globe équipé du réseau Internet. La stratégie de sauvegarde journalière interdit les transactions de plus d'une journée. Nous présenterons dans la partie V les pistes que nous envisageons d'explorer pour améliorer la gestion de la persistance, en exploitant par exemple les mécanismes avancés de base de données pour la coopération que nous avons étudiés au chapitre cinq.

CHAPITRE 14

LES POINTS DE VUE TECHNOLOGIQUES

1. LES SERVEURS CO-LEARN

Les serveurs sont installés sur des stations de travail exploités sous UNIX. L'environnement de développement et d'exécution Smalltalk d'ObjectWorks tourne dans l'environnement X-Windows. La base de données nécessite au minimum 200 MO d'espace disque.

En ce qui concerne la mémoire vive, il faut prévoir 2 MO pour le système d'exploitation UNIX, 2 MO pour l'environnement X, 2 MO pour le moniteur de la base d'objets GemStone, 3 MO pour chaque session GemStone, 4 MO pour le tampon de fichier du système d'exploitation. La couche de communication de Co-Learn nécessite la capacité de supporter de nombreux processus simultanés et de nombreux descripteurs de fichiers ouverts représentant les sockets. Nous recommandons au total un minimum de 16 MO. Actuellement, nous utilisons des stations de travail de milieu de gamme, dotées de 24 MO de mémoire.

Il faut noter que le serveur Co-Learn peut être porté sur toute machine sur laquelle tourne Smalltalk-80 et qui supporte le logiciel de réseau TCP/IP [STEVE 90] : seule la couche de communication nécessiterait éventuellement d'être modifiée. Par exemple, comme l'environnement Smalltalk fonctionne également sous Windows, il est envisagé de porter le serveur sur Windows NT, dans le cadre du projet européen DEMOS en cours de soumission dans le programme DELTA.

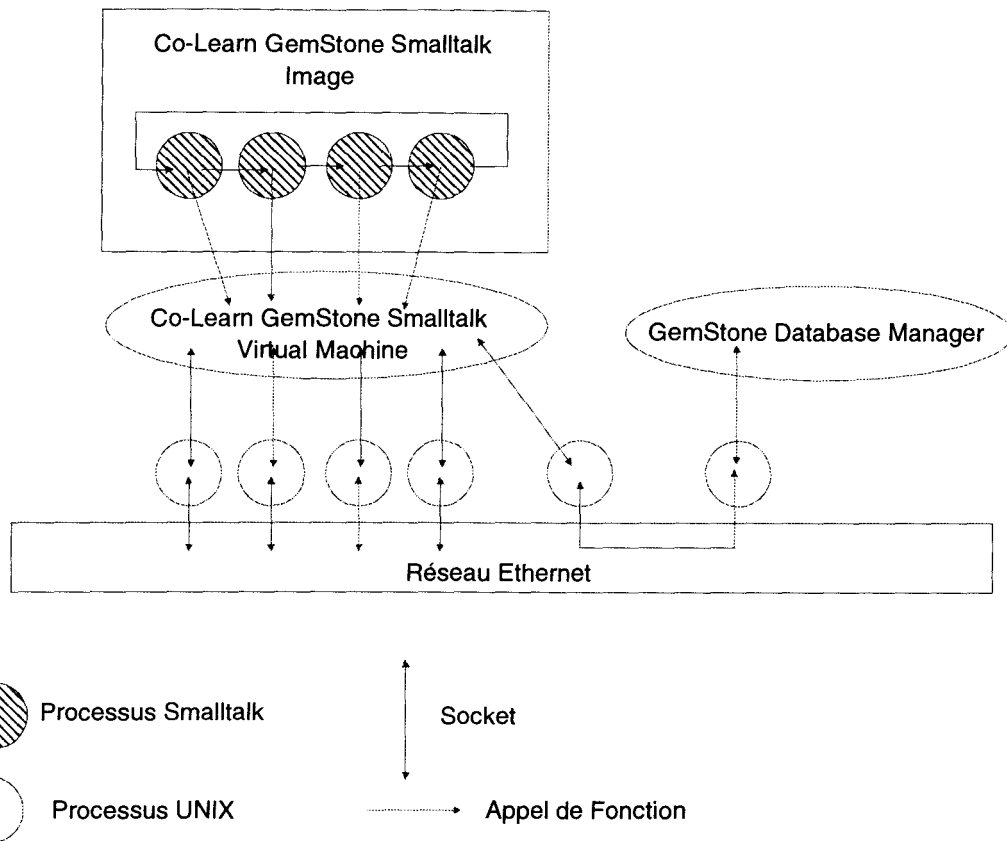


Figure 71. Les composants technologiques d'un serveur Co-Learn

2. LES CLIENTS CO-LEARN

Les clients Co-Learn sont des PC multimédia exploités sous Windows 3.1, équipés d'un microphone et de haut-parleurs ou d'un casque ainsi qu'un écran VGA ou Super VGA. Nous avons réalisé une classification des stations clientes Co-Learn en fonction de leur équipement et du mode d'accès :

A : accès au serveur Co-Learn par l'intermédiaire d'une carte Ethernet (et du réseau Internet)

B : accès au serveur Co-Learn par l'intermédiaire d'une carte et du réseau RNIS

C : accès au serveur Co-Learn par l'intermédiaire d'un modem et du Réseau Téléphonique Commuté

1 : accès au pont audio avec un téléphone extérieur

2 : accès au pont audio directement par le PC

S : le PC dispose d'une carte son aux normes MPC qui fournit des possibilités de connectique avec la carte RNIS. Ceci signifie que l'on peut contrôler le son par l'intermédiaire d'outils informatiques.

Une classe de machine est donc obtenue à partir du langage élémentaire suivant :

$$(A|B|C)(1|2)[S]$$

3. LE RESEAU CO-LEARN

Les serveurs Co-Learn existants actuellement sont installés sur les réseaux Ethernet de l'Université de Lille et de l'université d'Aveiro, d'autre part.. De ce fait, toute machine de ces universités peut entrer en contact avec le serveur Co-Learn par Ethernet. De plus, le réseau de l'université est relié via Noropale au réseau Internet. Toute machine connectée à l'Internet peut donc contacter le serveur Co-Learn.

Plusieurs postes clients Co-Learn du CUEEP sont connectés au réseau Ethernet de l'université. Se situant sur le même brin Ethernet, ils bénéficient d'un débit important pour la communication des données, bien qu'il ne soit pas garanti. Lorsqu'ils veulent participer à une audioconférence, les utilisateurs de ces postes doivent utiliser un téléphone pour contacter le pont audio, installé sur le Réseau Téléphonique Commuté.

Les postes clients Co-Learn sont plus souvent connectés au réseau RNIS. Ils accèdent au serveur Co-Learn par l'intermédiaire du routeur IP que nous avons installé sur le réseau de l'université. Ils bénéficient d'un débit qui a l'avantage d'être garanti constant, mais l'inconvénient de n'être que de 64 Kb, ce qui est généralement plus faible que sur réseau Ethernet. L'avantage de cette configuration est l'intégration dans le PC Multimédia de la communication sonore, ce qui permet de régler finement le confort d'écoute et d'assurer automatiquement l'établissement des communications avec le pont audio pour la gestion des conférences.

Claude Viéville a récemment vérifié le fonctionnement et l'utilisabilité des clients Co-Learn connectés par le Réseau Téléphonique Commuté, par l'utilisation d'un modem et du protocole de lien PPP. Ceci permet d'augmenter l'accessibilité du système. Un poste sur RTC permet d'exploiter les services de coopération en temps différé de Co-Learn. Il est également possible d'utiliser les possibilités de coopération en temps réel, avec certaines contraintes : le temps de téléchargement des documents étant fortement augmenté, il faut prévoir un chargement préalable à la session de travail des documents sur le poste client et éviter l'utilisation du collage public.

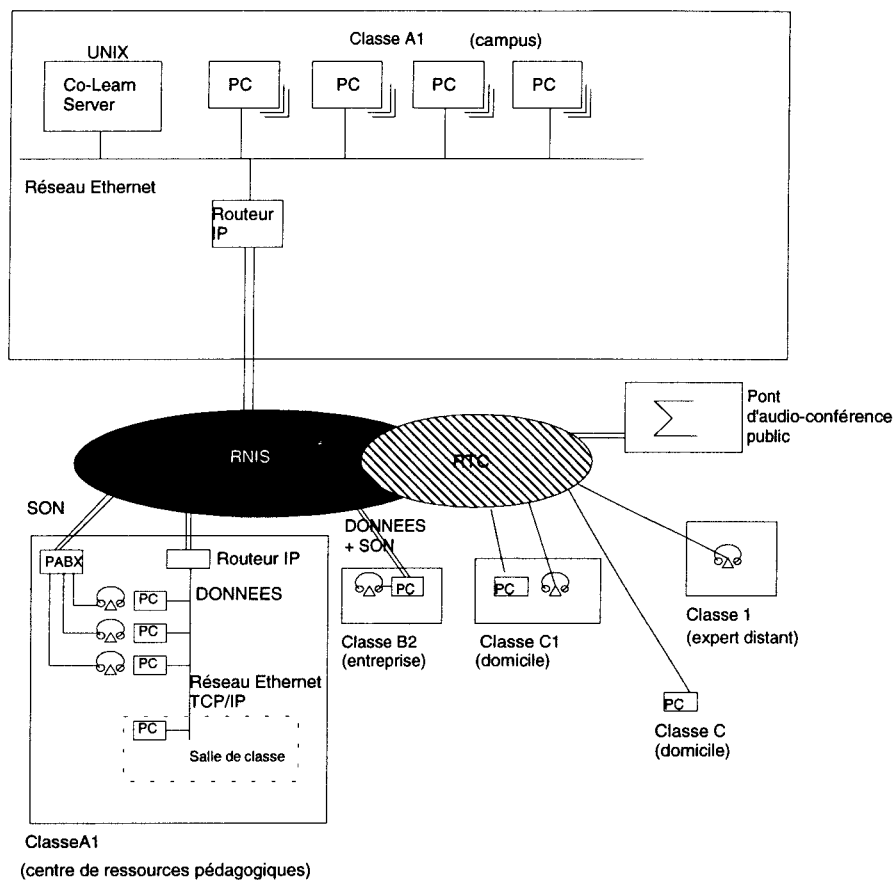


Figure 72. Le réseau physique Co-Learn

Nous étudions la possibilité d'optimiser l'accès au service Co-Learn afin de pouvoir proposer un accès public de type kiosque [VIEVI 94a]. Dans un premier temps, nous distribuons les services de communication à l'aide de relais pour optimiser la diffusion sur le réseau [LUQUE 95]. Dans un deuxième temps, nous prévoyons de distribuer le serveur au plus près des points d'accès aux services, à l'instar du réseau télématique. Pour ceci, plusieurs études doivent être menées : le portage du serveur Co-Learn sur Smalltalk distribué, l'utilisation de bases GemStone fédérées. La principale difficulté prévue est la répercussion de la distribution des objets Smalltalk sur la transparence de la gestion de la persistance. Nous nous intéressons donc aux travaux de recherche menés dans le domaine des systèmes distribués à objets persistants tels que PVC-BOX [NAMYS 93].

**PARTIE V
EVALUATION ET AMELIORATION DE CO-LEARN**

TABLE DES MATIERES DE LA CINQUIEME PARTIE

CHAPITRE 15 EVALUATION DE CO-LEARN ET PERSPECTIVES D'EVOLUTION	242
1. De l'ACAO au TCAO : développer le support des activités instrumentales	242
1.1. Co-Learn favorise les activités de communication par rapport aux activités instrumentales	243
1.2. Articulation des espaces d'information et de communication dans Co-Learn	243
1.3. Supporter l'édition coopérative avec Co-Learn	249
2. augmenter la malléabilité de Co-Learn	254
3. augmenter l'accessibilité du système Co-Learn.....	255
3.1. Améliorer les services de communication de groupe	255
3.2. Multiserveur, distribution et objets nomades.....	255
4. Mieux assister l'orientation dans l'environnement de coopération.....	256
5. Améliorer la gestion de la persistance.....	256

CHAPITRE 15

EVALUATION DE CO-LEARN ET PERSPECTIVES D'EVOLUTION

1. DE L'ACAO AU TCAO : DEVELOPPER LE SUPPORT DES ACTIVITES INSTRUMENTALES

Co-Learn n'est pas réellement un système de TCAO. Il est spécialisé pour l'apprentissage coopératif : c'est donc un système d'ACAO¹ (Apprentissage Coopératif Assisté par Ordinateur, CSCL en Anglais : Computer Supported Cooperative Learning). Dans son état actuel, le système privilégie fortement l'aspect communicationnel de la coopération parce que la **communication** est traditionnellement l'activité essentielle de l'apprentissage. Le travail coopératif n'est pas une fin en soi : les apprenants n'ont pas pour objectif de mener à bien une tâche. Ils doivent tirer des leçons des échecs et succès de l'activité menée pour tenter d'accomplir la tâche, prétexte de l'apprentissage. En mode synchrone comme en mode asynchrone, Co-Learn permet de discuter "autour" de documents. Co-Learn supporte bien les activités de conversation mais pour faire de Co-Learn un système de travail coopératif plus complet permettant la **production** de groupe, il faudrait développer le support des **activités instrumentales**. La participation à des activités instrumentales pourrait être utile aussi en situation d'apprentissage, pour permettre aux apprenants de développer un savoir-faire.

¹A ne pas confondre avec le système ACAO d'Audio-Conférence Assisté par Ordinateur conçu par Laurent Barme pour étudier le rôle du canal sonore dans le TCAO [BARME 92][BARME 94a][BARME 94b][BARME 95].

1.1. Co-Learn favorise les activités de communication par rapport aux activités instrumentales

Dans les activités de communication, les documents sont utilisés comme un support à la conversation. Les participants n'agissent pas sur le contenu des documents mais au-dessus d'eux (par l'annotation) ou autour d'eux (par la désignation). Ce qui compte, c'est ce que les apprenants disent, et leurs gestes autour de l'espace de visualisation de l'information visent à renforcer, illustrer leur discours. Bien sûr, le sujet de la conversation peut être une action : on parle de ce que l'on pourrait faire, de ce que l'on devrait faire, de ce que l'on a fait ou de ce que l'on fera mais on ne travaille pas effectivement en présence du groupe. Dans une activité instrumentale, au contraire, l'action est centrale et la communication éventuelle sert à améliorer l'action par la coordination. Les activités instrumentales que l'informatique peut supporter sont surtout des activités de manipulation d'information. Le type d'activité instrumentale que l'informatique supporte le plus naturellement est l'édition coopérative de documents structurés ou de réseaux hypermédia.

1.2. Articulation des espaces d'information et de communication dans Co-Learn

Le principe général de Co-Learn est de permettre de développer un espace de communication et un espace d'information et de tisser des liens entre ces deux espaces.

La coopération est gérée dans l'espace de communication mais pas dans l'espace d'information. Co-Learn gère le partage de la base de documents mais pas le partage des documents.

1.2.1. Le télé-enseignement

L'outil de télé présentation de Co-Learn utilisé dans les Salles Virtuelles de cours magistral permet l'exploration commentée en groupe d'un document hypermédia préalablement produit avec l'éditeur. Les participants partagent un espace de visualisation, un espace de désignation et un espace de communication. Il n'est pas prévu de communication entre les espaces d'information commun et privé. Il n'y a ni espace d'annotation, ni espace d'édition.

1.2.2. La téléconférence

Dans une téléconférence, les participants peuvent produire individuellement des documents hypermédia et les enregistrer dans la bibliothèque de l'organisation.

Les auteurs peuvent signaler par un message qu'ils ont produit un nouveau document ou modifié un document existant. Pour faciliter l'accès à ce document, ils créent un lien du message vers une ancre du document hypermédia. Ce document peut être une partie du dossier dont la réalisation constitue la tâche du groupe. Ce peut être aussi le compte-rendu détaillé d'une activité instrumentale externe au système. Les participants partagent un espace de communication, un espace d'information et un espace de désignation. Ils accèdent individuellement à un espace d'édition privé et à l'espace d'information partagé.

1.2.3. La télé réunion

Le besoin de modification coopérative des éléments d'information est illustré par les demandes exprimées par les utilisateurs et les évaluateurs d'une amélioration des fonctionnalités du télépointeur. Ils n'arrivent pas à accepter que le télépointeur ne soit qu'un outil de **désignation**, de support de la gestuelle. Ainsi, les marques du télépointeur sont pour eux des **annotations** et ils regrettent qu'on ne puisse pas les manipuler comme des objets graphiques d'un éditeur de dessin. Les utilisateurs demandent de pouvoir construire avec le télépointeur des objets graphiques plus évolués. Mais surtout, ils ne comprennent pas que les schémas qu'ils réalisent avec le télépointeur ne puissent pas être enregistrés dans le document partagé. Les utilisateurs ont besoin de manipuler un artefact commun. Cette constatation rejoint l'idée exprimée dans [SCHWA 94] selon laquelle il est nécessaire de faire travailler les participants d'une réunion sur des artefacts communs pour conserver une mémoire organisationnelle et permettre *moins de bavardage et plus de travail (less talk and more work)*.

Sous la pression des utilisateurs, Pascal Croisy a ajouté à la fenêtre publique la possibilité d'enregistrer dans un nouveau document le contenu du document affiché avec les marques de désignation. Cependant le système incite fortement les utilisateurs à enregistrer le document annoté dans leur espace privé. Le télépointeur et la fenêtre publique conçus comme des outils d'aide à la **communication** et à la **confrontation** de points de vue sont ainsi devenus des moyens d'**annotation** coopérative. Cette transformation impulsée par les utilisateurs permet à un participant de conserver dans son espace privé une trace des conversations en temps réel. Cependant, cette modification provoque une rupture par rapport aux outils de coopération en temps différé et ne reflète pas toujours fidèlement l'activité essentiellement communicationnelle qui a eu lieu durant la session en temps réel. Il est probable que cette modification ne satisfasse pas encore certains utilisateurs qui voient dans la combinaison de la fenêtre publique et du télépointeur un seul outil d'**édition** coopérative. En fait, la fenêtre publique est vue à la fois comme :

- un **espace de partage d'information** : elle est d'abord conçue pour visualiser un extrait de l'espace d'information commun autour duquel on va construire la conversation,

- un **espace de communication** : on peut l'utiliser pour transmettre des messages à l'attention du groupe. Un exemple courant est la présentation en début de session d'un document présentant les instructions pour la tâche à mener. Ces instructions seront en général énoncées oralement mais leur présentation écrite permet d'en améliorer la compréhension et la mémorisation. Les utilisateurs peuvent en faire une copie dans leur espace privé pour les reprendre quand il le juge utile. Le groupe sait qu'il pourra aussi les consulter à nouveau au cours de la session. La fenêtre publique adjoint aux avantages de la communication orale les qualités de communication liées à l'écrit (en outre la révisabilité).

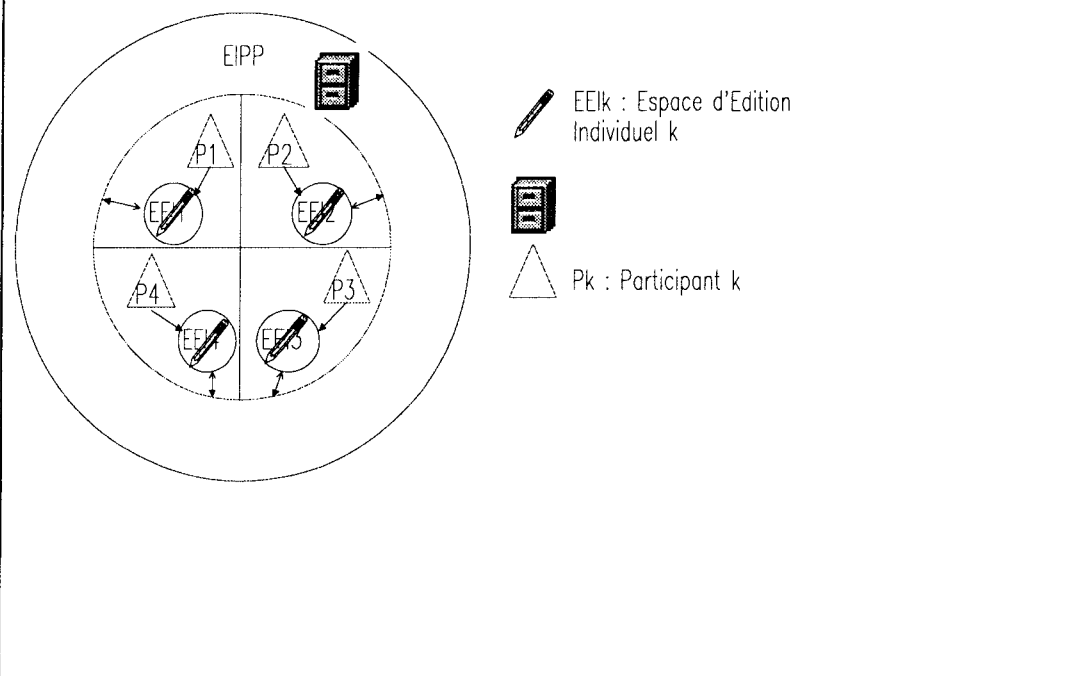
- un **espace de production** : en offrant les possibilités de transfert entre l'espace de production privé et la fenêtre publique, le système ressemble, à une échelle de temps restreinte, à l'édition coopérative en **mode séparé**.

Au cours d'une session Co-Learn d'apprentissage coopératif en temps réel, les utilisateurs peuvent utiliser les outils d'édition de leur espace privé pour construire des documents, qui sont par exemple des versions de leur compréhension du problème. A certains instants, les participants peuvent porter leur travail à la connaissance de tous, à l'aide de la fenêtre publique. Les participants coopèrent donc dans une situation très semblable au mode d'accès séparé à une base d'information que supporte un éditeur hypermédia de groupe tel que Cooperative SEPIA. La **confusion** de ces deux situations est à l'origine de leur déception. Dans le mode séparé d'un éditeur hypermédia coopératif, il y a **enrichissement de l'espace d'information persistant commun** alors que dans la situation de coopération en temps réel supporté par Co-Learn, il y a uniquement **enrichissement des espaces d'information privés**² et enrichissement d'un espace d'information partagé volatile (l'espace de visualisation).

D'un certain point de vue, il s'agit en fait de deux situations symétriques par rapport à un plan de symétrie qui séparerait l'espace d'information de l'espace de communication. Dans le cas de l'utilisation de l'hypermédia coopératif en mode séparé, l'utilisateur se situe dans un espace où il travaille seul et l'espace partagé persistant est l'espace extérieur vers lequel il envoie accessoirement et par intermittence les résultats intermédiaires significatifs de son travail (figure 73). Il ne communique pas avec les autres participants.

²Il est possible pour un expert en informatique d'ajouter les documents qu'il réalise au cours d'une session de coopération en temps réel dans l'espace d'information partagé de la Salle Virtuelle, mais il s'agit alors d'une déviation du système.

Une situation de coopération orientée production d'information en mode séparé.
L'espace de production individuel est central.
L'espace d'information partagé est périphérique
Chaque participant travaille dans son espace d'édition individuel pour modifier l'espace d'information partagé persistant



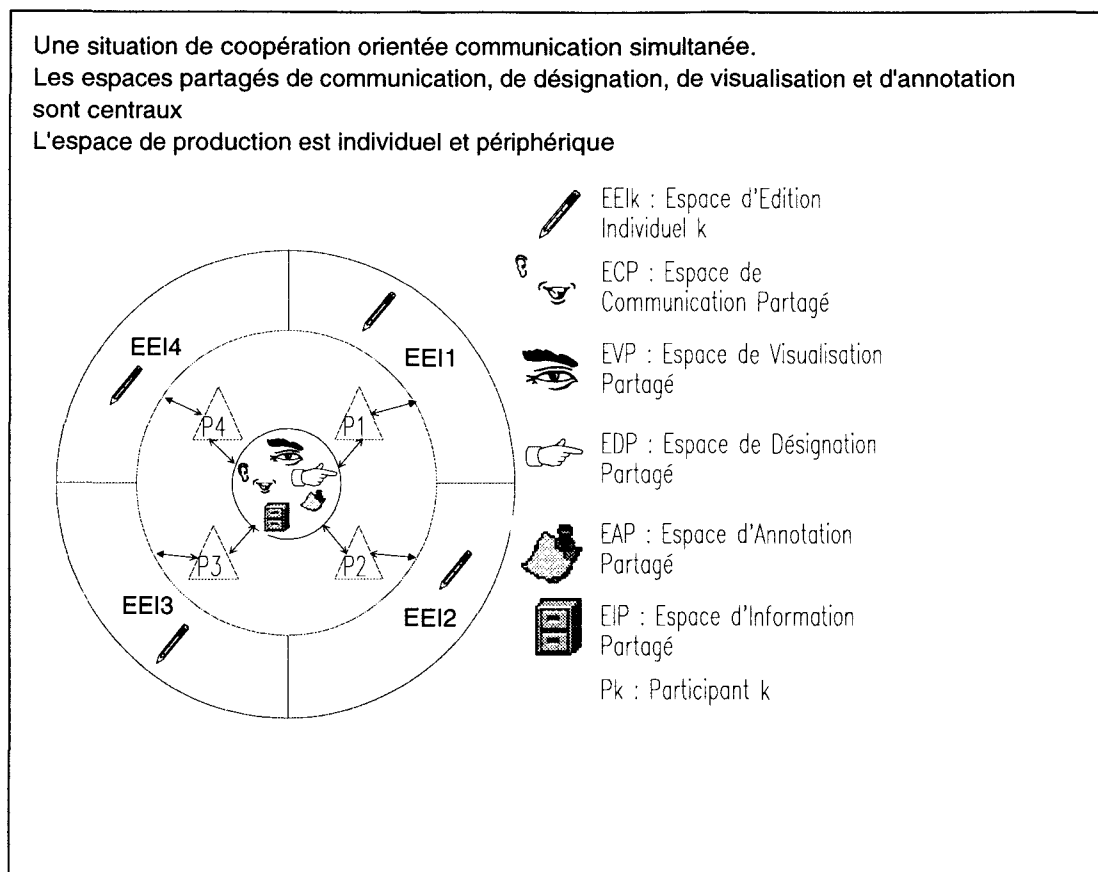


Figure 74. articulation des espaces dans une situation Co-Learn de réunion en temps réel

1.2.4. La télé assistance

Dans la situation de télé assistance, deux personnes peuvent coopérer en situation fortement couplée. En utilisant un outil d'édition, les participants peuvent dévier le système pour coopérer à la production d'un document. Ils partagent alors un espace d'action commun complet, incluant les espaces de communication, d'information, de visualisation, de désignation, d'annotation et d'édition. Cependant, il ne s'agit pas d'une édition de groupe, puisqu'elle est limitée à deux personnes.

1.2.5. Accès à l'espace d'information partagé

1.2.5.1. Accès en consultation

Actuellement, les utilisateurs accèdent à la base d'information grâce au gestionnaire de leçons qui permet de retrouver des documents par leur nom ou par la recherche de mots clés que l'auteur du document peut leur associer. L'espace d'information devrait être structuré pour enrichir les moyens d'accès direct. Les documents contenus dans la base sont des documents hypermédia, qui permettent à l'utilisateur une exploration par suivi de liens. Mais les documents hypermédia n'ont pas de liens entre eux, qui permettraient de naviguer dans la base d'information globale.

1.2.5.2. Accès en modification

La modification d'un élément de l'espace d'information (i.e. un document hypermédia) n'est possible qu'en mode isolé : seul l'auteur qui l'a créé peut modifier dans son espace privé un document hypermédia de la librairie. Pour supporter la production de groupe, il faudrait ajouter des moyens de construction coopérative de documents hypermédia. Aujourd'hui, les utilisateurs qui veulent coopérer à la construction d'un espace d'information ne peuvent se tenir informés que par l'intermédiaire des outils de conversation. Ceci signifie que la rétroaction doit être gérée explicitement par les utilisateurs dans l'espace de conversation en temps différé. De ce fait, la rétroaction est elle-même toujours différée.

1.3. Supporter l'édition coopérative avec Co-Learn

Une nouvelle version de Co-Learn, si elle devait supporter la production de groupe, devrait sans doute proposer un outil d'édition coopérative. Il serait intéressant de pouvoir utiliser cet outil dans les différentes situations d'édition identifiées par [HAAKE 92].

1.3.1. Situations actuelles d'édition dans Co-Learn

Actuellement, l'environnement Co-Learn permet d'éditer un document dans différentes circonstances mais toujours en mode isolé.

Un utilisateur peut créer un document alors qu'il se trouve dans la librairie virtuelle, il utilise alors ses outils d'édition habituels (éditeurs de texte, de dessins, de graphiques, tableurs, enregistreurs de sons...) pour créer les objets multimédia, puis l'éditeur hypermédia pour les intégrer et enfin le gestionnaire de leçons pour les stocker dans la librairie partagée. Un utilisateur qui se trouve dans la librairie peut également utiliser le gestionnaire de leçons pour récupérer un document hypermédia et le stocker à nouveau après modification, à condition d'en avoir le droit. Par l'utilisation de verrous applicatifs, le système assure que personne d'autre ne modifie le document, c'est à dire que l'éditeur travaille en mode isolé.

Un utilisateur peut créer un document pour le référencer dans un message d'une conversation en temps différé. Par exemple, il peut s'agir du compte-rendu d'une action dans le cadre d'une conversation pour l'action ou de l'ordre du jour ou des minutes d'une réunion dans le cadre d'une conversation pour l'organisation de réunions. Ici encore, l'utilisateur travaille en mode isolé.

Enfin, un utilisateur peut éditer un document au cours d'un atelier de travail de groupe. Il utilise alors ses outils habituels pour produire le document, dans son espace privé. Il ne peut pas utiliser la fenêtre publique pour éditer le document : il ne peut l'utiliser que pour montrer au groupe le résultat de son travail. Tous les participants peuvent alors désigner le document pour en discuter. Le groupe peut également annoter le document et chacun pourra recopier le document dans son espace privé, éventuellement avec les annotations que le groupe a apportées à l'aide du télépointeur. Mais la fenêtre publique n'est pas conçue pour produire un document en groupe. Eventuellement, un utilisateur pourra utiliser les copies qu'il aura faites des documents avec les annotations pour produire un nouveau document, ou pour rédiger un compte-rendu de la réunion, qu'il pourra stocker dans la librairie et éventuellement référencer dans un message d'une conversation en temps différé.

1.3.2. Publication de l'information produite

De même que Co-Learn permet de produire des documents dans différentes circonstances, il est possible de présenter ces documents au groupe à différentes occasions. Un utilisateur peut présenter un document en l'attachant à un message dans une conversation en temps différé. Le même document peut être présenté à un groupe au cours d'une session de télé-enseignement (ou plus généralement de télé-présentation). Des pages extraites du document peuvent aussi être présentés à l'aide de la fenêtre publique dans une Salle Virtuelle d'atelier de travail. Enfin, un utilisateur peut consulter le document dans la librairie virtuelle.

1.3.3. Co-Learn favorise l'autonomie coopérative

Ainsi, dans Co-Learn, le travail est finalement individuel. Co-Learn est un très bon outil pour l'apprentissage coopératif à distance dans le contexte de l'enseignement ouvert. Il préserve l'autonomie des apprenants tout en favorisant la coopération car il incite à discuter en groupe du travail réalisé ou à réaliser et il oblige chaque apprenant à utiliser seul les outils pour produire un travail personnel qui contribuera au travail du groupe. En cas de difficulté au cours de son travail individuel, l'apprenant peut toutefois obtenir l'aide d'un technicien, d'un expert ou d'un formateur, par exemple en entrant dans une Salle Virtuelle de télé assistance qui permet le travail coopératif en situation fortement couplée avec l'assistant. C'est ainsi que Co-Learn concilie les enjeux de cohésion et d'implication. Il permet la construction sociale du savoir par la confrontation des points de vue autour du travail réalisé par des apprenants autonomes.

On serait tenté de conclure que tant que Co-Learn est utilisé pour l'ACAO (Apprentissage Coopératif Assisté par Ordinateur), il ne paraît pas nécessaire de permettre l'édition coopérative de document. Cette conclusion peut être contestée dans le cas où l'on veut utiliser Co-Learn pour apprendre l'édition coopérative. Le support d'activités coopératives en temps réel autre que l'édition coopérative peut aussi être nécessaire dans certains cas particuliers. Par exemple, l'apprentissage de la méthode de planification KANBAN peut être réalisé par une simulation d'entreprise où chaque apprenant est responsable d'un poste [VILER 92]. Faut-il considérer qu'une telle application est spécifique et doit être développée de façon autonome de l'environnement Co-Learn, sans utiliser ses mécanismes de coordination et de communication de groupe ? Doit-on offrir des mécanismes de base facilitant son intégration ? Doit-on faciliter le développement d'une telle application dans l'environnement Co-Learn ? Qu'est-ce qu'il faudrait changer à Co-Learn pour cela ?

L'intégration d'un mécanisme tel que SharedX semble intéressante pour permettre à un groupe de partager l'utilisation de n'importe quel outil. Malheureusement, il ne semble pas possible de développer un mécanisme équivalent dans l'environnement Windows.

L'auteur peut choisir le mode différé pour informer les utilisateurs de son action. Il lui appartient alors de créer un message dans un échange d'une tâche de la salle virtuelle appropriée. L'avantage par rapport à un hypermédia partagé est la possibilité d'expliquer les modifications apportées et de les situer dans le contexte d'un échange, d'une tâche et d'un espace de coopération (Salle Virtuelle). L'inconvénient est qu'on ne supporte pas les situations de coopération couplée (simultanée), sauf dans les situations de télé assistance mais elles ne permettent de coopérer qu'à deux participants.

1.3.4. Difficulté de "coopérisation" de l'éditeur Hypermédia de Co-Learn

L'éditeur de document hypermédia³ [MARTI 94] [BORST 94] [PINTO 94] tel qu'il est proposé actuellement dans Co-Learn supporte les activités instrumentales individuelles, mais pas les activités d'édition partagée. L'intérêt majeur de cet éditeur est l'ouverture qu'il offre grâce à l'utilisation de la technologie OLE ®. Il permet ainsi d'intégrer les documents réalisés par les participants avec l'éditeur de leur choix, contribuant ainsi à l'enjeu d'implication. Mais l'utilisation de la technologie OLE est aussi à l'origine du principal inconvénient de l'éditeur. En effet, l'OLE ayant pour effet de cacher la structure des objets OLE qui composent un document, le contrôle du partage ne peut pas s'opérer sur un objet plus fin que l'objet OLE complet, qui est un gros objet puisqu'il peut être un texte, un dessin, un tableau, un graphique une séquence vidéo ou sonore.

Currently all OLE servers are (to the best of our knowledge) single user applications. This implies that co-authoring in OLE based hypermedia editors is restricted to the level of composing hypermedia pages in the form of cooperative re-arranging of the graphical presentations of OLE objects.

Due to the fact that the format OLE objects are transparent for the archive, write and read administration can only take place on object level. This restrict us in abilities, to differentiate between parts of an OLE object with or without write protection.

[BORST 94]

1.3.5. Concevoir et intégrer des activités Co-Learn d'édition coopérative

Nous étudions actuellement avec Pascal Croisy la possibilité de concevoir et d'intégrer dans l'architecture Co-Learn des activités d'édition de groupe de documents structurés telle que celle supportée par Alliance [DECOU 92] [DECOU 94].

³Cet outil a été conçu et développé par nos partenaires de l'INESC de l'Université d'Aveiro (Portugal), qui l'ont baptisé HyDE.

Nous voyons deux possibilités pour concevoir les activités d'édition coopérative de document structuré.

La première méthode consiste à instancier une activité de coopération en temps réel pour l'édition de chaque fragment. Le rôle d'un participant dans une activité définit donc l'évolution de ses droits sur le fragment, en fonction de l'état du fragment (traduit par la phase de l'activité) et du mode d'édition dans lequel le participant accède actuellement au fragment (traduit par l'état du rôle).

La deuxième méthode consiste à associer une activité de "conversation" par édition de document entier. L'objectif de la conversation est alors l'édition d'un document structuré. Les personnes impliquées dans la conversation ne manipulent pas des messages mais des fragments. Les rôles ne définissent pas les droits de réponse à un message mais les droits d'accès (lecture, édition, gestion des rôles) à un type de fragment en fonction de son état (libre, édité, géré), de la phase de l'activité d'édition du document et de l'état du rôle dans l'édition du document. Ainsi les droits d'accès d'un coauteur aux différents types de fragments du document changent lorsque l'activité d'édition globale du document évolue.

La première solution offre les mêmes fonctions qu'un système d'édition coopérative tel que Alliance. L'avantage est la flexibilité du système puisque le gestionnaire d'un fragment est totalement libre d'affecter les droits d'accès au fragment à qui bon lui semble. L'avantage de la seconde solution est de permettre de définir pour un type de document donné une politique générale d'affectation et d'évolution automatique des rôles sur les différents types de fragment du document. Ceci permet de coordonner l'édition des différents fragments. On peut par exemple assister la production des documents dans le cycle de développement d'un logiciel, en précisant les types de fragment (spécification, programme source, documentation utilisateur, documentation technique, documentation commerciale, information de maintenance...), les rôles impliqués dans la manipulation de l'information (prescripteur, analyste, concepteur, programmeur, ergonomes, évaluateur, utilisateur...) et l'évolution de leurs droits (lecture, édition, annotation, gestion, création de version...) des différents types de fragments en fonction de la phase de l'activité (appel d'offre, analyse, conception, réalisation, évaluation, distribution, maintenance...).

Intégration du World Wide Web dans l'environnement de coopération

Il est possible d'intégrer WWW à différents niveaux dans l'environnement Co-Learn.

Le degré élémentaire d'intégration consiste à mémoriser uniquement des URLs (Uniform Resource Locator), par exemple pour attacher ces références à des messages de conversation en temps différé. Ceci permet à un groupe de partager une base de références de documents accessibles par le réseau Internet.

Le deuxième niveau d'intégration est obtenu en permettant l'exploration partagée du Web. Cette intégration est possible en utilisant les interfaces DDE et OLE offertes par le visualiseur Netscape [BLYTH 95a] [BLYTH 95b]. En effet, il est possible d'établir une conversation DDE avec Netscape au cours de laquelle Netscape communique toutes ses actions. Netscape peut donc être intégré à Co-Learn comme un outil partagé. Une activité sur le serveur peut aisément prendre en charge la synchronisation des visualiseurs des participants. Cependant, il ne serait pas possible de gérer des rôles dans cette activité d'exploration partagée du Web car le protocole DDE proposé par Netscape ne permet pas à notre connaissance de modifier les ressources d'actions de l'interface.

Un troisième niveau d'intégration devrait permettre l'édition coopérative de documents sur le Web. Cependant, nous risquons de retrouver les mêmes difficultés de partage que celles que nous rencontrons avec l'éditeur Hypermédia liée à l'encapsulation des objets composant le document.

2. AUGMENTER LA MALLEABILITE DE CO-LEARN

Actuellement, les services de gestion disponibles sur les postes clients Co-Learn ne permettent de définir que des rôles structurels. Ils ne proposent pas la gestion des rôles fonctionnels, qui sont par conséquent attribués par défaut par le système en fonction du rôle structurel dans la Salle Virtuelle. Pour modifier la distribution des rôles, un administrateur de Salle Virtuelle doit demander l'intervention d'un administrateur du serveur, à l'aide de formulaires prévus à cet effet. Nous envisageons de développer un protocole et un outil de gestion des rôles fonctionnels dans une salle. Les méthodes sont déjà disponibles sur le serveur, définissant du fait le protocole d'interaction : il n'y a donc plus qu'à développer les modules clients conformément à ce protocole.

3. AUGMENTER L'ACCESSIBILITE DU SYSTEME CO-LEARN

3.1. Améliorer les services de communication de groupe

Comme nous l'avons vu dans le chapitre 13, il est nécessaire de modifier l'interface entre le serveur et la couche de communication, pour éviter la scrutation inutile des différents processus de communication attachés aux salles virtuelles.

Par ailleurs, la diffusion des messages est réalisée par un processus central par salle virtuelle. Cette solution présente l'avantage de garantir facilement l'atomicité de la diffusion : tous les utilisateurs reçoivent les messages diffusés dans le même ordre. Cependant, en cas d'utilisation intensive de la diffusion dans une salle, le processus central pourrait constituer un goulot d'étranglement pénalisant tous les participants présents dans la même salle. C'est pourquoi, dans le cadre du projet OSACA de Ganymède, nous envisageons l'utilisation d'une couche de communication de groupe telle que ISIS.

Son application à Co-Learn implique des contraintes sur la qualité des services, pour supporter les flux multimédia.

Un autre défaut actuel de Co-Learn est l'absence d'une couche session qui permette la reprise en cas de coupure du circuit virtuel de communication d'une salle virtuelle. L'ajout d'une couche session au niveau de la salle virtuelle est donc un des objectifs du laboratoire.

3.2. Multiserveur, distribution et objets nomades

Dans la perspective d'une extension du réseau Co-Learn à une large échelle, nous envisageons la distribution et la fédération des serveurs, comme nous l'avons vu dans le chapitre 14. Nous nous intéressons dans ce cadre aux travaux du projet COMIC, en particulier [SARMI 93]. La distribution d'un serveur consiste à répartir les objets d'une même organisation au plus près des points d'accès au service. La fédération des serveurs a pour but de mettre en relation différentes organisations.

D'autre part, pour augmenter la disponibilité du système, nous avons commencé à étudier avec l'équipe RD2P (Réseau Distribué et Dossier Portable) du Laboratoire d'Informatique Fondamentale de Lille les conséquences de l'installation des clients Co-Learn sur des objets nomades.

4. MIEUX ASSISTER L'ORIENTATION DANS L'ENVIRONNEMENT DE COOPERATION

Actuellement, le seul moyen d'orientation dans l'environnement de coopération est la structuration spatiale de l'univers des Salles Virtuelles de l'organisation. Lorsqu'un participant se connecte à Co-Learn, il ne reçoit aucune aide pour choisir le cours et la Salle Virtuelle dans laquelle il serait intéressant qu'il se rende, en fonction du temps. L'évaluation de Co-Learn [COLEARN 95] a imputé ce manque de prise en compte de la dimension temporelle à la métaphore du centre de ressources virtuel. Selon [COLEARN 95], lorsque l'utilisateur se connecte, la question qu'il se pose et à laquelle le système devrait lui répondre est "qu'est ce que je devrais faire maintenant", plutôt que "où suis-je sensé me trouver ?". Notre opinion est que la métaphore de la Salle Virtuelle est très utile pour assister l'organisation d'environnement de coopération et proposer des contextes aux activités des participants mais n'est pas le bon moyen d'orientation des utilisateurs. Ceci correspond à la situation réelle, où l'utilisateur consulte son agenda pour savoir ce qu'il doit ou peut faire à un moment donné, au lieu d'explorer l'ensemble des lieux où il est autorisé à se rendre. Or le système possède des renseignements qui peuvent aider l'utilisateur à se situer sur l'axe temporel des activités, par exemple :

- les prochaines réunions auxquelles l'utilisateur doit participer,
- les salles virtuelles dans lesquelles de nouvelles réponses ont été données aux messages de l'utilisateur,
- les salles virtuelles dans lesquelles une réponse de l'utilisateur est attendue,
- les salles de télé assistance dans lesquelles un tuteur est présent...

Nous prévoyons donc de modifier le logiciel d'accueil de Co-Learn pour offrir à l'utilisateur une aide à l'organisation. La métaphore de l'Agenda, proposée par Carmel Smith, semble bien adaptée et va être évaluée.

5. AMELIORER LA GESTION DE LA PERSISTANCE

Actuellement, le serveur ne gère qu'une seule transaction, partagée par tous les utilisateurs et toutes les applications Co-Learn. L'avantage de cette approche est la facilité de gestion de la coopération et de la rétroaction de groupe. L'inconvénient est que le travail réalisé par un groupe ou un individu dans une Salle Virtuelle ne peut pas être enregistré séparément. La transaction n'est enregistrée que lorsque tous les utilisateurs sont déconnectés.

Une première solution est de conserver un seul environnement Smalltalk (éventuellement distribué) partagé par l'ensemble des applications et utilisateurs d'une organisation Co-Learn mais de gérer dans l'environnement Smalltalk plusieurs sessions en parallèles. L'interdépendance de l'ensemble des objets du système d'information rend cependant difficile son découpage en unités atomiques cohérentes. La meilleure candidate à ce rôle de délimitation d'unité cohérente est la Salle Virtuelle. L'idée est d'attacher une transaction à chaque Salle Virtuelle : nous pouvons ainsi gérer une transaction par groupe de travail, au lieu d'une transaction pour l'ensemble de l'organisation. Les objets modifiés dans une Salle Virtuelle (Rôles, Activités, Messages, Echanges, Tâches, Participants) sont enregistrés à la sortie du dernier utilisateur de la Salle Virtuelle. Ainsi, après une session de travail en groupe, tous les utilisateurs quittent la pièce et leur travail est sauvegardé immédiatement. De même, quand un utilisateur isolé se connecte pour effectuer un travail asynchrone de courte durée, il est probable qu'il reste seul dans la Salle Virtuelle le temps de sa connexion et par conséquent son travail sera enregistré dès son départ de la Salle.

Cette solution pose encore problème pour les Salles Virtuelles dans lesquelles les participants restent connectés longtemps ou se relayent, empêchant l'enregistrement des objets de la Salle. Il faudrait dans ce cas déterminer des unités de cohérence plus fines que la Salle Virtuelle. Comme les requêtes adressées au serveur sont sérialisées, une solution envisageable consiste à enregistrer la transaction après chaque exécution de requête amenant le système d'objets à un état stable. Dans ce cas, pour éviter la sauvegarde du système dans un état transitoire, toute activité doit pouvoir détecter l'incohérence générée par l'exécution d'une requête et interdire tout enregistrement de la transaction jusqu'à réception d'une requête rétablissant la cohérence. Le risque de report permanent d'enregistrement des objets diminue avec cette solution mais ne peut pas être écarté.

En fait, un système d'objets partagé par un ensemble de collaborateurs est pratiquement toujours dans un état transitoire. Ainsi, il faut soit se résoudre à enregistrer ces états transitoires instables, soit déterminer et n'enregistrer que les parties autonomes du système qui sont un dans état stable. Ceci nous rappelle l'utilisation du mécanisme de check-in/check-out proposé avec les transactions longues. Cependant, ce mécanisme a pour objet de publier les objets à destination des autres transactions (et donc de relaxer le principe d'isolation) alors que dans notre cas nous nous affranchissons du principe d'isolation en partageant la transaction et nous utilisons l'enregistrement uniquement pour assurer la durabilité du travail du groupe. La transaction telle que nous envisageons de l'utiliser n'aurait pas pour rôle d'isoler les utilisateurs mais de partitionner le système d'objets global de l'organisation en unités autonomes.

Les méthodes les plus avancées proposées dans la communauté des bases de données pour tenter de répondre aux besoins des applications complexes supportées par les bases de données à objets combinent les transactions ouvertes et imbriquées. Comme [OZSU 93] l'explique, ces transactions ne sont plus des transactions car elles ne respectent pas les principes ACID. En effet, les transactions ouvertes ne respectent pas le principe d'Atomicité car elles autorisent l'enregistrement partiel des objets manipulés dans la transaction. Elles ne respectent pas non plus le principe de Durabilité car elles autorisent à annuler les effets d'une transaction enregistrée, à l'aide d'une transaction de compensation. [OZSU 93] propose d'appeler ces fausses transactions des activités ou *Workflows*.

En conclusion, les transactions ACID ne sont pas adaptées aux activités complexes supportées par les bases de données orientées objets, et en particulier les activités de coopération. L'interdépendance des objets d'un système d'information pour le travail coopératif impose l'utilisation de transactions partagées par des groupes d'utilisateurs. La gestion de ces transactions nécessite l'implication des objets applicatifs qui ont seuls les connaissances nécessaires à la détection des états stables du système d'information partagé. La transaction ne doit plus être utilisée pour isoler le travail d'un utilisateur mais pour isoler un ensemble autonome d'objets manipulés par un groupe d'utilisateurs.

CONCLUSION

Ces travaux sur la conception des environnements informatiques d'aide au travail coopératif appliqués à l'éducation ont permis de participer à la conception, la réalisation et l'évaluation en situation réelle de Co-Learn, une plate-forme distribuée multimédia d'apprentissage coopératif à distance.

L'identification de six enjeux du travail de groupe et l'estimation de leur satisfaction par les différents types de collectif montrent l'intérêt d'un environnement intégré d'aide à la coopération.

Les travaux de cette thèse ont conduit à la définition d'une architecture d'accueil distribuée orientée objet et au développement d'une méthode de conception multi-agent des activités de coopération en temps réel et en temps différé des groupes organisés. L'architecture ouverte offre des services de communication de groupe, des mécanismes de construction, d'organisation et de gestion d'environnements de coopération persistants dans lesquels les utilisateurs jouent des rôles dans des activités de coopération à l'aide d'outils partagés. L'environnement est capable d'intégrer toute application existante, à un degré qui ne dépend que du degré d'ouverture de l'application en terme de moyens standards de coopération entre applications (DDE, OLE). La méthode de conception d'activités de coopération, dérivée du modèle AMIGO, est intégrée dans une méthode globale de conception par points de vue d'un environnement de TCAO (CSCW) articulée autour de la métaphore de la Salle Virtuelle.

Aujourd'hui, les activités intégrées à la plate-forme sont essentiellement des activités de communication de groupe en temps réel et en temps différé adaptées à l'apprentissage coopératif à distance. Un des objectifs sur lequel l'auteur souhaite continuer ses travaux est l'étude de l'adéquation ou de l'adaptabilité de l'architecture et de la méthode de conception aux activités instrumentales des organisations, telles que l'édition coopérative ou la circulation de documents.

D'autre part, comme l'évaluation en situation réelle l'a montré, il est crucial de mieux intégrer dans Co-Learn la dimension temporelle pour assister les utilisateurs dans l'organisation de leur travail, la gestion de leur temps de travail.

Un autre but est d'adapter l'architecture au support du travail coopératif à grande échelle, par la distribution des serveurs et la fédération des plates-formes des organisations.

Enfin, l'amélioration de la gestion de la persistance des objets du système d'information pour le travail coopératif implique une nouvelle approche des mécanismes transactionnels pour les bases de données à objets.

REFERENCES BIBLIOGRAPHIQUES

[AGOST 94] Alessandra AGOSTINI, Giorgio DE MICHELIS, Stefano PATRIARCA, Renat TININI : "**A Prototype of an Integrated Coordination Support System**", Computer Supported Cooperative Work (CSCW), Chapitre 2, pp 209-238, Kluwer Academic Publishers, Pays Bas, 24 Janvier 1994.

[AHMED 90] Shamim AHMED, Duvvuru SRIRAM, Robert LOGCHER : "**A Comparison of Object-Oriented Database Management Systems for Engineering Applications**", Technical Report N° IESL-90-03, Department of Civil Engineering, Intelligent Engineering Systems Laboratory, Massachusetts Institute of Technology, Cambridge, 11 Octobre 1990.

[ANDLE 92] Prabhat K. ANDLEIGH, Michael R. GRETZINGER : "**Distributed Object-Oriented Data-Systems Design**", chapitre 14, section 14.4, Prentice hall, 1992, pp 358-359.

[ANZIE 82] D ANZIEU, J.-Y. MARTIN : "**La dynamique des groupes restreints**", puf, le psychologue, Presses Universitaires de France, première édition en mai 1968, dernière édition en décembre 1982.

[AURAM 92a] E. AURAMÄKI, R. HIRSCHHEIM, K. LYYTINEN : "**Modelling Offices Through Discourse Analysis : The SAMPO Approach**", in The Computer Journal, vol. 35, n° 4, 1992.

[AURAM 92b] E. AURAMÄKI, R. HIRSCHHEIM, K. LYYTINEN : "**Modelling Offices Through Discourse Analysis : A Comparison and Evaluation of SAMPO with OSSAD and ICN**", in The Computer Journal, vol. 35, n° 5, 1992.

[AUSTI 62] J. L. AUSTIN : "**Quand dire c'est faire**", Le Seuil, 1962.

[BARME 92] L. BARME, A. DERYCKE : "**Intégration du son dans les applications du collecticiel**", in [TELEC 92].

[BARME 94a] L. BARME : "**Le canal sonore dans le CSCW**", in [GANYM 94].

[BARME 94b] L. BARME : "**Audio-Conférences Assistées par Ordinateur : gestion et contrôle**", in [TRIGO 94], pp. 133-138.

[BARME 95] L. BARME : "**Structure d'un collecticiel : un système multi-agents**", in [GANYM 95], pp. 89-92

[BEAUD 92] M. BEAUDOIN-LAFFON, A. KARSENTY : "**Transparency and Awreness in a Real-Time Groupware System**", in Proc. ACM Symposium on User Interface Software and Technology (UIST), Novembre 1992.

[BENFO 91] S. BENFORD : "**A Real World Metaphor for a CSCW Environment**", CoTech WG2 Internal Document, 1991.

[BENFO 92] S. BENFORD, M. TUROFF, J. PALME : "**An ISO standard to support asynchronous group communication**" in Computer Standards & Interfaces 14 (1992), pp 363-373, Elsevier Science Publishers B.V., 1992.

[BENFO 93] S. BENFORD, W. PRINZ, J. MARIANI, T. RODDEN, L. NAVARRO, E. BIGNOLI, C. G. BROWN, T. NASLUND : "**MOCCA - A Distributed Environment for Collaboration**", Actes de Telepresence'93, "First International Conference in Technologies and Theories for Human Cooperation, Collaboration, Coordination", APPLICA'93, Lille, France, 22-24 Mars 1993, pp. 135-148.

[BENTL 94] Richard BENTLY, Tom RODDEN, Pete SAWYER, Ian SOMMERVILLE : "**Architectural Support for Cooperative Multiuser Interfaces**", in COMPUTER, Computer Supported Cooperative Work, Mai 1994.

[BEVER 93] M. BEVER, K. GEIHS, L. HEUSER, M. MUHLHÄUSER, A. SCHILL : "**Distributed Systems, OSF DCE, and Beyond**", IN DCE - The OSF Distributed Computing Environment. Client/Server Model and Beyond, Springer-Verlag, Actes de International DCE Workshop, Karlsruhe, Allemagne, Octobre 1993, pp. 1-20.

[BIRMA 90] K. BIRMAN, R. COOPER, T. JOSEPH, K. MARZULLO, M. MAKPANGOU, K. KANE, F. SCHMUCK, M. WOOD : "**The ISIS System Manual. technical report**", The ISIS project, Cornell University, Septembre 1990.

[BLYTH 95a] Garrett Arch BLYTHE : "**Netscape's DDE Implementation. Version 0x00010000**", 22 Mars 1995.

URL : <http://home.netscape.com/newsret/std/ddeapi.html>.

[BLYTH 95b] Garrett Arch BLYTHE : "**OLE Automation in Netscape**", 22 Mars 1995.

URL : <http://home.netscape.com/newsret/std/oleapi.html>.

- [**BOGIA 92**] Douglas P. BOGIA, William J. OLONE, Simon M. KAPLAN : "**Modeling Dynamic Obligations in ConversationBuilder**", Human-Computer Interaction Laboratory, University of Illinois, 12 Août 1992.
- [**BORST 94**] H.W.J. BORST PAUWELS, J. SOUSA PINTO, B. SOUSA SANTOS, J.A. MARTINS : "**Integrating Existing Applications in Hypermedia Learning Material (General Issues and Experiences with OLE Technology)**", in EDMEDIA'94, pp. 444-449.
- [**BOWER 88**] J. BOWERS, J. CHURCHER, T. ROBERTS : "**Structuring Computer-Mediated Communication in COSMOS**", in R. SPETH (Ed) : "Research into Networks and Distributed Applications", EUTECO 88, North Holland Publishers, 1988.
- [**BREWE 90**] John BREWER, Debra A. WALKER : "**Guide de l'utilisateur Timbuktu**", Farallon Computing, Inc., 1990.
- [**BUTTER 91**] Paul BUTTERWORTH, Allen OTIS, Jacob STEIN : "**THE GEMSTONE OBJECT DATABASE MANAGEMENT SYSTEM**", in Communications of the ACM, Volume 34, N°10, Octobre 1991, pp. 64-77.
- [**CARD 87**] Stuart K. CARD, D. Austin HENDERSON, Jr. : "**Catalogues : A Metaphor for Computer Application Delivery**", in Human Computer Interaction - INTERACT'87, H.-J. Bullinger, B. Shackel (editors), Isevier Science Publishers (North Holland), IFIP, 1987.
- [**CARD 91**] Stuart K. CARD, George G. ROBERTSON, Jack D. MACKINLEY : "**The Information Visualizer, An Information Workspace**", in ACM/CHI/91, pp. 181-188.
- [**CARDO 93**] E. CARDOZO, J.S. SICHMAN, Y. DEMEZEAU : "**Using the Active Object Model to Implement multi-Agent Systems**", Actes de 1993 IEEE Int.'l Conf. on Tools with AI, Boston, Massachussets, Novembre 1993.
- [**COLEARN 92a**] Open University, Trigone, Polytechnical University of Madrid : "**Development Methods, Standards and Norms**", DELTA Project D2005 COLEARN Deliverable 1, Mai 1992.
- [**COLEARN 92b**] Open University, Polytechnical University of Madrid, Télésystèmes, Trigone : "**Representation models for collaborative educational situations and collaborative learning activities**", DELTA Project D2005 COLEARN Deliverable 4, Juin 1992.

[COLEARN 92c] Télésystèmes, Polytechnical University of Madrid, Trigone : **'Specification of the common object oriented toolbox'**, DELTA Project D2005 CO-LEARN Deliverable 3, Juin 1992.

[COLEARN 92d] Trigone : **"Specification of the asynchronous Computer Mediated Communication system"**, DELTA Project D2005 CO-LEARN Deliverable 11, Septembre 1992.

[COLEARN 92e] Technische Hochschule Darmstadt, Polytechnical University of Madrid, Télésystèmes : **"Definition of Co-Learn General Architecture"**, DELTA Project D2005 CO-LEARN Deliverable 14, Décembre 1992.

[COLEARN 92f] Trigone : **"Software of the common object-oriented toolbox"**, DELTA Project D2005 CO-LEARN Deliverable 15, Décembre 1992.

[COLEARN 93a] Polytechnical University of Madrid (DIT-UPM) : **"Reference Manual of Communication Services"**, DELTA Project D2005 CO-LEARN Deliverable 18B, Octobre 1993.

[COLEARN 93b] Software de Base S.A. : **"First CPI version"**, DELTA Project D2005 CO-LEARN Deliverable 19, Septembre 1993.

[COLEARN 93c] Technische Hochschule Darmstadt : **"Programmers Reference Manual for Multimedia Object Update & Retrieval"**, DELTA Project D2005 CO-LEARN Deliverable 20B, Septembre 1993.

[COLEARN 93d] M. SALL : **"User Manual of the Real-Time Tele-Assistance Tool"**, DELTA Project D2005 CO-LEARN Deliverable 22B, Septembre 1993.

[COLEARN 93e] C. VIEVILLE : **"User Manual of the Asynchronous Multimedia Conferencing System"**, DELTA Project D2005 CO-LEARN Deliverable 23B, Septembre 1993.

[COLEARN 93f] Pascal CROISY, Claude VIEVILLE : **"User manual of real-time multimedia conference"**, DELTA Project D2005 CO-LEARN Deliverable 24, Octobre 1993.

[COLEARN 94a] M. SALL : **"User Manual for the Real Time Tele-Teaching Tool"**, DELTA Project D2005 CO-LEARN Deliverable 21B, Mars 1994.

[COLEARN 94b] Joaquim Arnaldo MARTINS, Joaquim Sousa. PINTO, Hubert W.J. BORST PAUWELS, Beatriz Sousa SANTOS : **"CPI - Second Version. Hypermedia Document Editor and Hypermedia Document Manager"**, DELTA Project D2005 CO-LEARN Deliverable 27-2, Juillet 1994.

- [COLEARN 95] UK Open University (Tony KAYE) : "**FINAL EVALUATION REPORT. Synthesis of Co-Learn trials and Experimentation**", DELTA Project D2005 CO-LEARN Deliverable 33, Mars 1995.
- [COMIC 93] L. BANNON, K. SCHMIDT : "**Issues of supporting Organizational in CSCW Systems**", Deliverable D1.1, COMIC esprit basic research project 6225, 1er octobre 1993.
- [CROIS 95a] Pascal CROISY : "**Collecticiel temps réel et apprentissage coopératif: des aspects sociaux et pédagogiques jusqu'au modèle multi-agent de l'interface de groupe**", Thèse de Doctorat mention Informatique soutenue à l'Université des Sciences et Technologie de Lille en Février 1995.
- [CROIS 95b] Pascal CROISY : "**Un modèle multi-agent pour l'interface de groupe**", in [GANYM 95], pp. 47-66 .
- [DANIE 89] Thore DANIELSEN : "**AAM - The AMIGO Activity Model**", in [PANKO 89], pp. 67-125.
- [DANIELS 93] John DANIELS, Steve COOK : "**Strategies for sharing objects in distributed systems**", Journal of Object Oriented Programming, vol. 5, N° 8, janvier 1993.
- [DARCH 94] Jean-Claude DARCHEVILLE : "**Mais qu'est-ce que la coopération ?**", in [GANYM 94].
- [DECOU 86] D. DECOUCHANT : "**Design of a distributed Object Manager for the Smalltalk-80 System**", ACM OOPSLA'86, Septembre 1986, pp. 444-452.
- [DECOU 92] D. DECOUCHANT, V. QUINT, I. VATTON : "**L'édition coopérative de documents avec Griffon**", in [TELEC 92], pp. 137-141.
- [DECOU 94] Dominique DECOUCHANT : "**Rétroaction de groupe et édition coopérative de documents structurés**", in [TRIGO 94], pp. 145-150.
- [DELGA 91] Jaime DELGADO, Manuel MEDINA : "**Group Communication Mailbox Server Accessible through a Standard Mailbox Client**", in Message Handling Systems and Applications Layer Communication Protocols, P. Schicker and E. Stefferud (Eds), Elsevier Science Publishers B.V. (North Holland), IFIP, 1991, pp 177-189.

[DERYC 90] Alain DERYCKE, Claude VIEVILLE, Pierre VILERS : "**Cooperation and communication in Open Learning : the CoConut project**", in Computer in Education, A.Lc Dougall and C. Dowling (editors), North-Holland, Proceedings of the WCC'90 IFIP Conference, Sidney, pp 956-962, juillet 1990.

[DERYC 92] A. DERYCKE, D. CLEMENT, C. LADESOU, C. VIEVILLE : "**Towards a hypermedium for collaborative learning**", in A.R. KAYE (Ed) Collaborative Learning through Computer Conferencing : The Najaden papers", NATO ASI Series, Vol. 90, Berlin, Springer-Verlag, 1992, pp. 211-224.

[DERYC 93] A. DERYCKE : "**Apprentissage ouvert et à distance : vers un processus coopératif impliquant de nouveaux partenariats et de nouveaux rôles pour les auteurs**", Journées Internationales de l'IDATE, Montpellier, France, Novembre 1993.

[DERYC 94] A. DERYCKE : "**Les difficultés du travail coopératif : un point de vue individuel**", in [GANYM 94].

[DERYC 95] Alain DERYCKE, Frédéric HOOGSTOEL : "**Le travail coopératif assisté par ordinateur : quels enjeux pour les concepteurs**", INFORSID'95, à paraître, 1995

[DEUX 89] O. DEUX et al. : "**The Story of O₂**", rapport technique 37-89, GIP Altaïr, Rocquencourt, France, 14 octobre 1989. In IEEE Transactions on Knowledge and Data Engineering, Volume 2, Number 1, Mars 1990.

[DIGIT 94] Digital Equipment Corporation : "**LinkWorks. Manuel d'utilisation**", janvier 1994.

[DICEN 90] G. DICENET : "**LE RNIS : Techniques et Atouts**", Collection Technique et Scientifique des Télécommunications, MASSON, 1990.

[DOBSO 91] J.E. DOBSON, M.J. MARTIN, C.W. OLPHERT, S.E. POWRIE : "**Determining Requirements for CSCW : the ORDIT Approach**", in Collaborative Work, Social Communications and Information Systems, R.K. STAMPER, P. KEROLA, R. LEE and K. LYYTINEN (Editors), Elsevier Science Publishers B.V. (North-Holland), IFIP, 1991.

- [DOGAC 93] Asuman DOGAC, M. Tamer ÖZSU, Alexandros BILIRIS, Timos SELLIS (Eds) : "**Advances in Object-Oriented Database Systems**", Proceedings of the NATO Advanced Study on Object-Oriented Database Systems, Izmir, Kusadasi, Turkey, August 6-16, 1993, NATO ASI Series, Series F : Computer and Systems Sciences, Vol. 130.
- [DOLLI 91] J. DOLLIMORE, E. MIRANDA, WANG XU : "**The Design of a System for Distributing Shared Objects**", in THE COMPUTER JOURNAL, Vol. 34, n° 6, 1991.
- [DUGER 91] Philippe DUGERDIL : "**Smalltalk-80. Programmation par objets**", Collection Informatique, Presses Polytechniques et Universitaires Romandes, 1991.
- [ECL 93] Ecole Centrale de Lyon (Eds) : "**Actes des cinquièmes journées sur l'ingénierie des Interfaces Homme-Machine IHM'93**", Lyon, Octobre 1993.
- [ELLIS 94] Clarence Skip ELLIS, Jacques WAINER : "**A Conceptual Model for Groupware**", CSCW 94 - 10/94 Chapel Hill, NC, USA, 1994.
- [ENGEL 92] Douglas C. ENGELBART : "**Toward high-performance organizations : a strategic role for groupware**", in CSCW 92 Proceedings, pp. 77-100.
- [ENGES 87] Y. ENGESTROM : "**Learning by expanding**", Orienta-konsultit, Helsinki, 1987.
- [FLORE 86] F. FLORES, T. WINOGRAD : "**Understanding computer and cognition : new foundations for design**", Ablex Publishing, Norwood, New Jersey, 1986.
- [FLORE 88] Fernando FLORES, Michael GRAVES, Brad HARTFIELD, Terry WINOGRAD : "**Computer Systems and the design of Organizational Interaction**", in ACM transactions on Office Information Systems, Vol. 6, N° 2, April 1988, pp. 153-172.
- [GANYM 94] Document de travail, "**Workshop Ganymède, Travail Coopératif et Communication Avancée**", Lille, France, 3 juin 1994.
- [GANYM 95] Document de travail, "**Workshop Ganymède, Travail Coopératif et Communication Avancée. Actes des journées de travail sur la notion de Multi-Agents**", Publication n° 163, LIFL USTL, Lille, 17 Février 1995.
- [GOLDB 83] A. GOLDBERG, D. ROBSON : "**Smalltalk-80 : The Language and its Implementation**", Addison-Wesley Inc., 1983.

- [GRIET 91] J.J. van GRIETHUYSEN : "**Enterprise modelling, a necessary basis for modern information systems**", in Open Distributed Processing, J. De Meer, V. Heymer, R. Roth (Editors), Elsevier Science Publishers B.V. (North Holland), Actes de IFIP TC6/WF6.4 International Workshop on Open Distributed Processing, Berlin, Allemagne, 8-11 Octobre 1991, pp 29-68.
- [GROCH 82] E. GROCHLA : "**Grundlagen der organisatorischen Gestaltung**", Sammlung Poeschel, C.E. Poeschel Verlag Stuttgart, 1982.
- [GRØNB 94] Kaj GRØNBAEK, JENS A. HEM, OLE L. MDASEN, LENNERT SLOTH : "**SYSTEMS : A DEXTER-BASED ARCHITECTURE**", in Communication of the ACM, Volume 37, N° 2, Février 1994, pp 65-74.
- [GUST 88] P. GUST : "**SharedX : X in a shared group environment**", in the second annual Technical Conference, Massachusetts Institute of Technology, USA, 1988
- [HAAKE 92] Jörg M. HAAKE, Brian WILSON : "**Supporting Collaborative Writing of Hyperdocuments in SEPIA**", Actes de CSCW'92, pp 138-146, Novembre 1992.
- [HAAKE 93] Anja HAAKE, Jörg M. HAAKE : "**Take CoVer : Exploiting Version Support in Cooperative Systems**", Actes de INTERCHI'93, pp. 406-413, 24-29 Avril 1993.
- [HAMMA 87] Heikki HÄMMÄINEN, Reijo SULONEN : "**Pages : Intelligent Forms, Intelligent Mail and Distribution**", in OFFICES SYSTEMS : Methods and Tools, G. Bracchi, D. Tschritzis (Editors), Elsevier Science Publishers (North Holland), IFIP, 1987.
- [HAMMA 91] Heikki HÄMMÄINEN, Chris CONDON : "**Form and room : Metaphors for Groupware**", COSC 91/ACM, pp 95-104, 1991.
- [HELLER 92] Martin HELLER (Ed) : "**Advanced Windows Programming**", WILEY Professional Computing, 1992.
- [HENDE 86] D. A. HENDERSON, Jr, S.K. CARD : "**ROOMS : The use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface**", in ACM Transactions On Graphics, Volume 5, N° 3, juillet 1986, pp 211-243.

- [HENNE 94] Pippa HENNESSY, Paul HARVEY, Hugh SMITH : **"Support for enterprise modelling for CSCW"**, in Collaborative Computing 1, 1994, pp. 127-146.
- [HOOGS 92a] F. HOOGSTOEL, C. VIEVILLE, E. PASTOR : **"The AMIGO Approach : the Information Model"**, in [COLEARN 92b]; pp. 59-89.
- [HOOGS 92b] Frédéric HOOGSTOEL : **"Au croisement de trois interfaces ... l'assistant d'animation : un agent informatique au coeur d'un collecticiel"**, in [TELEC 92].
- [HOOGS 93] Frédéric HOOGSTOEL, Claude VIEVILLE : **"Conception et réalisation d'un système d'intégration des activités d'apprentissage coopératif"**, in [ECL 93], pp 73-80.
- [HOOGS 94] Frédéric HOOGSTOEL : **"L'intégration d'un dispositif informatique pour le travail coopératif dans une organisation"**, in [GANYM 94].
- [HUYTO 92] Alan HUYTON, DELTA (Ed) : **"Telematics Systems for Flexible and Distance Learning DELTA'92"**, Mars 1992.
- [INSOF 95] Insoft : **"Communique ! Insoft Desktop Conferencing And Workgroup Collaboration. User's Guide"**, 20 janvier 1995.
- [IRIS 89] E. IRIS : **"Le RNIS. Présentation et applications pour l'utilisateur"**, DUNOD Informatique, 1989.
- [ISHII 91] H. HISHII, N. MIYAKE : **"Toward an Open Shared Workspace : Computer and Video Fusion Approach of TeamWorkStation"**, Communications of the ACM, Vol. 34, N° 12, Décembre 1991.
- [ITASC 90] **"ITASCA Distributed Object Database Management System Technical Summary"**, ITASCA SYSTEMS, INC., 1990.
- [JOHNS 91] Peter JOHNSON-LENZ, Trudy JOHNSON-LENZ : **"Post-mechanistic groupware primitives : rhythms, boundaries and containers"**, in Saul Greenberg (ed) : **"Computer-Supported Cooperative Work and Groupware"**, publié chez ACADEMIC PRESS, 1991.
- [KARSE 94] Alain KARSENTY : **"Le collecticiel : de l'interaction homme-machine à la communication homme-machine-homme"**, in "Technique et Sciences Informatiques", numéro spécial, "Interface Homme-Machine", volume 13, n°1, 1994, pp. 105-127.

- [KAYE 92] A.R. KAYE (Ed) : "**Collaborative Learning Through Computer Conferencing**", NATO ASI Series, Springer Verlag, 1992
- [KUUTT 91] Kari KUUTTI : "**The concept of Activity as a Basic Unit for CSCW Research**", in L.J. BANNON, M. ROBINSON, K. SCHMIDT (eds) : "Proceedings of the second ECSCW. Kluwer Academical, Amsterdam, pp. 249-264, 1991.
- [KUUTT 93] Kari KUUTTI : "**Notes on systems supporting *organizational context* - an activity theory viewpoint**", in [COMIC 93].
- [LEPOU 93] Patrick LEPOUTRE : "**Théories de la conversation et application dans Co-Learn**", Mémoire de D.E.A. d'Informatique, laboratoire Trigone, L.I.F.L., Lille, 1993.
- [LI 91] Q. LI, D. McLEOD : "**An Object-Oriented Approach to Federated Databases**", Proceedings of First International Workshop on Interoperability of Multidatabases Systems, mars 1991, pp. 64-70.
- [LININ 91] P. F. LININGTON : "**Introduction to the Open Distributed Processing Basic Reference Model**", in Open Distributed Processing, J. De Meer, V. Heymer, R. Roth (Editors), Elsevier Science Publishers B.V. (North Holland), Actes de IFIP TC6/WF6.4 International Workshop on Open Distributed Processing, Berlin, Allemagne, 8-11 Octobre 1991, pp 3-13.
- [LUQUE 95] Samuel LUQUET : "**Création de concentrateurs de réseau et extension a des fonctions relais**", manuel de réalisation, rapport de projet de 3ème année, EUDIL, département IMA.
- [MADSE 89a] C. M. MADSEN : "**Using Persistent Objects to Implement an Environment for Cooperative Work**", in TOOLS'89, Septembre 1989, pp 243-252.
- [MADSE 89b] C. M. MADSEN : "**Approaching Group Communication by Means of an Office Building Metaphor**", in First European Conference on Computer Supported Cooperative Work, Septembre 1989, pp 449 - 460.
- [MAIER 90] D. MAIER, J. STEIN : "**Development and Implementation of an Object-Oriented DBMS**", in Stanley B. ZDONIZ, David MAIER : "Readings in OBJECT-ORIENTED DATABASE SYSTEMS", 1990, pp. 167-185.
- [MALON 87] T. W. MALONE, K. R. GRANT, K. Y. LAI, R. RAO, D. A. ROSENBLITT : "**Semi-structured messages are surprisingly useful for computer-supported coordination**", ACM Transactions on Office Information Systems, 1987, volume 5, pp 115-131.

[MALON 90] Thomas W. MALONE, Kevin CROWSTON : "**What is Coordination Theory and How Can It Help Design Cooperative Work Systems ?**", in CSCW'90 Proceedings, octobre 1990, pp. 357-370.

[MALON 92] Thomas W. MALONE, Kum-Yew LAI : "**Experiments with Oval: A Radically Tailorable Tool for Cooperative Work**", MIT, Mars 1992.

[MANTY 90] Risto MÄNTYLÄ, Jari ALASUVANTO, Heikki HÄMMÄINEN : "**Pages : a Testbed for Groupware Applications**", in Multi-User Interfaces and Applications, S. Gibbs, A. A. Verijn-Stuart (Editors), Elsevier Science Publishers (North Holland), IFIP, 1990, pp 37-47.

[MARMO 91] Hans MARMOLIN, Peter LIDBAUM, Yngve SUNDBLAD, Magnus TOBIASSON : "**The Collaborative Desktop : Interface and Functionalities**", in Actes de ...

[MARTI 94] Joaquim Arnaldo MARTINS, Joaquim Sousa PINTO : "**Hypermedia Courseware and Collaborative Tools for MS-Windows Environment**", in EDMEDIA'94, pp. 379-384

[MATHI 94] Philippe MATHIEU, "**Un système d'inférences multi-utilisateurs pour le travail coopératif**", in [GANYM 94].

[MOCCA 90] "**Working Group Proposal : A Model of an Environment for CSCW Applications**", CoTech WG2 Internal Document, 1990

[MOCCA 91] "**Models for CSCW Applications**", CoTech WG2 Internal Document, 1991.

[NAMYS 93] Raymond NAMYST : "**Projet PVC-BOX. Premiers pas vers une technologie de groupes**", mémoire de DEA, Laboratoire d'Informatique Fondamental de Lille, USTL, 7 décembre 1993.

[NAVAR 92] Leandro NAVARRO : "**An Environment for Open Group Cooperation**", in Open Distributed Processing, J. De Meer, V. Heymer, R. Roth (Editors), Elsevier Science Publishers B.V. (North Holland), 1992 IFIP, Actes de IFIP TC6/WF6.4 International Workshop on Open Distributed Processing, Berlin, Allemagne, 8-11 Octobre 1991.

[NICOL 94] J.C NICOLAS, B. CARRE, L. DBRAWER : "**Bases de Données Fédérées et Travail Coopératif**", in [GANYM 94].

[OZSU 93] M. Tamer ÖZSU : "**Transaction Models and Transaction Management in Object-Oriented Database Management Systems**", in [DOGAC 93], pp. 147-184.

[PALME 91] Jacob PALME : "**SuperCOM - a distributed computer conference system**", in Message Handling Systems and Applications Layer Communication Protocols, P. SCHicker and E. Stefferud (Eds), Elsevier Science Publishers B.V. (North Holland), IFIP, 1991, pp 191-200.

[PALME 92] Jacob PALME : "**Computer Conferencing Functions and Standards**", in [KAYE 92], pp. 225-245.

[PALME 93] Jacob PALME : "**Standards for asynchronous group communication**", in Computer Communications vol. 16, n° 9, Septembre 1993.

[PANKO 89a] Uta PANKOKE-BABATZ (Eds) : "**COMPUTER BASED GROUP COMMUNICATION MODEL. The AMIGO activity model**", Ellis Horwood Books in Information Technology, 1989.

[PANKO 89b] Uta PANKOKE-BABATZ : "**Group Communication and Electronic Communication**", in [PANKO 89a], pp. 17-66.

[PANKO 89c] Uta PANKOKE-BABATZ : "**Applications of AMIGO Work and Future Developments**", in [PANKO 89a], pp. 267-307.

[PETZO 92] Charles PETZOLD : "**Programmer sous Windows 3.1**", Microsoft Press, 1992.

[PINTO 94] J.S. PINTO, H.W.J. BORST PAUWELS, B. SOUSA SANTOS, J.A. MARTINS : "**HyDE : a hypermedia document editor based on OLE technology**", 1994 International Conference on Multimedia Computing and Systems, Boston, USA, 14-19 Mai 1994.

[PRINZ 89] Wolfgang PRINZ : "**Survey of Group Communication Models and Systems**", in [PANKO 89], pp. 128-180.

[PURDY 91] Alan PURDY, Bruce SCHUSCHERDT, David MAIER : "**Integrating an Object Server with Other Worlds**", in Rajiv GUPTA, Ellis HOROWITZ (Eds) : "Object Oriented database with Applications to CASE, Networks and VLSI CAD", pp. 216-236.

[RIFFL 95] Jean-Marie RIFFLET : "**La communication sous UNIX. Applications réparties**", Ediscience International, 1995.

[ROBER 90] Scott P. ROBERTSON, Wayne ZACHARY, John B. BLACK (Eds) : **"Cognition, Computing and Cooperation"**, publié par ALEX PUBLISHING CORPORATION, NORWOOD, NEW JERSEY, 1990.

[ROBIN 91] M. ROBINSON : **"Computer Supported Cooperative Work : Cases and Concepts"**, Groupware'91 - The Potential of Team and Organisational Computing, Chapitre 9, 1991, pp 59-75.

[RODDE 91] Tom RODDEN : **"A survey of CSCW systems"**, in Interacting with Computers, vol 3 n° 3, Butterworth-Heinemann Ltd, 1991.

[RONCA 94] Claudia RONCANCIO : **"Interopérabilité entre SGBD : systèmes fédérés et systèmes multibases"**, in Techniques et Sciences Informatiques, vol. 13, n° 3, 1994, pp 385-419.

[ROWE 91] Lawrence A. ROWE : **"A Shared Object Library"**, in On Object Oriented Database Systems, édité par Klaus R. Dittrich, Umeshwar Dayal, Alejandro P. Buchmann (Eds), Pringer-Verlag, 1991.

[SARMI 93] Mildred SARMIENTO, Leandro NAVARRO, Joan Manuel MARQUES : **"From Small to Large Scale"**, in [COMIC 93], pp. 141-158.

[SCHMI 90] Kjeld SCHMIDT : **"Cooperative work: a conceptual framework"**, in Rasmussen, Brehmer, Leplat (Eds) "Distributed decision making : cognitive model for cooperative work", Wiley, 1990.

[SCHMI 93] Kjeld SCHMIDT : **"Modes and Mechanisms of Interaction in Cooperative Work. Outline of a Conceptual Framework"**, Préparation du Rapport COMIC-Risø-3-3(2.0), Projet 6225 COMIC, Programme Européen ESPRIT, 1993.

[SCHMI 94] Kjeld SCHMIDT : **"COOPERATIVE WORK AND ITS ARTICULATION : REQUIREMENTS FOR COMPUTER SUPPORT"**, in "Le Travail Humain", tome 57, n°4, 1994, pp. 345-366.

[SEARL 82] SEARLE, **"Sens et expression : études des théories des actes de langage"**, Editions de minuit, 1982.

[SEGUI 92] J. SEGUIN, J. GUIVARCH, T. MORIN, Y. GUILLOT : **"Télé-Amphi : un service de communication de groupe"**, L'Echo de la Recherche, n°149, CNET, 1992.

[SERVI 91] Servio Corporation (Ed) : **"GemStone Product Overview"**, Février 1991.

[SHAPI 86] Marc SHAPIRO : "**Structure and Encapsulation in Distributed Systems : The Proxy Principle**", in Proc. 6th International Conference on Distributed Computer Systems, Mai 1986.

[SLAVI 85] R. SLAVIN, S. SHARAN, S. KAGAN, R.H. LAZAROWITZ, C. WEBB, R. SCHMUCK (Eds) : "**LEARNING TO COOPERATE, COOPERATING TO LEARN**", PLENUM PRESS, New York, USA, 1985.

[SMITH 89] Hugh SMITH, Julian ONIONS, Steve BENFORD (Eds): "**DISTRIBUTED GROUP COMMUNICATION. The AMIGO information model**", Ellis Horwood Books in Information Technology, 1989.

[SMITH 91] Hugh SMITH, Steve BENFORD, Alan Shepherd, Howidy HOWIDY : "**OSI-BASED GROUP COMMUNICATION**", in MESSAGE HANDLING SYSTEMS and Application Layer Communication Protocols, P. Schicker and E. Stefferud (Editors), Elsevier Science Publishers B.V. (North-Holland), IFIP, 1991.

[SMITH 94] C. SMITH, A.T. KAYE, J. HODGSON : "**Evaluating a Computer Supported Cooperative Learning System : An example : Co-Learn**", presentation between Heriot-Watt University and Université de Lille at a "CSCL" Workshop sponsored by CSCW SIG, 24 Mars 1994.

[STEVE 90] W. Richard STEVENS : "**UNIX Network Programming**", Prentice Hall Software Series, 1990

[STREI 94] Norbert A. STREITZ : "**Putting objects to work : Hypermedia as the Subject Matter and the Medium for Computer-Supported Cooperative Work**", in Object Oriented Programming, july 4-8,1994, pp 183-193.

[SUCHM 88] Lucy SUCHMAN : "**Designing with the User**", in ACM transactions on Office Information Systems, Vol. 6, N° 2, April 1988, pp. 173-183.

[SUNSO 92] SunSolutions (Ed) : "**ShowMe User's Guide**", Août 1992.

[TELEC 92] TELECOM Paris (Eds) : "**Document de travail des quatrièmes journées sur l'ingénierie des Intergacs Homme-Machine IHM'92**", décembre 1992, TELECOM Paris 92 S 004.

[TERRY 91] Douglas B. TERRY, "**7 steps to a better mail system**", in Message Handling Systems and Applications Layer Communication Protocols, P. Schicker and E. Stefferud (Eds), Elsevier Science Publishers B.V. (North Holland), IFIP, 1991.

- [TOUMA 93] F.TOUMANI, B. ERGIN, "**Validation collaborative des spécifications de systèmes d'information**", Actes du congrès INFORSID'93, Lille, 11-14 Mai 1993, édités par INFORSID, Juin 1993, pp 277-294.
- [TRIGO 94] Laboratoire TRIGONE (Eds) : "**Actes des sixièmes journées sur l'Ingénierie des Interfaces Homme-Machine IHM'94**", Lille, 8 et 9 décembre 1994.
- [VIEVI 90a] Claude VIEVILLE, Daniel POISSON, Alain DERYCKE : "**Global Network, Local and Regional, for Open Learning with CAI**", Seventh International Conference on Technology and Education, Bruxelles, Belgique, 20-22 Mars 1990, pp. 250-253.
- [VIEVI 90b] Claude VIEVILLE, Alain DERYCKE, Pierre VILERS : "**Architecture of a Collaborative System using Smalltalk and Unix**", in EUUG Spring'90 Proceedings, Munich, 23-27 Avril 1990, pp 89-98.
- [VIEVI 91] C. VIEVILLE, D. CLEMENT, R. DEMERVAL, A. DERYCKE, C LADESOS : "**To promote cooperative education in open learning by a dedicated computer mediated confernece and groupware**", in The 12th Educational Computing Organization on Ontario and the 8th International Conference on Technology on Education joint conference, Toronto, Canada , Mai 1991, pp 609-611.
- [VIEVI 92] Claude VIEVILLE, Frédéric HOOGSTOEL, Pascal CROISY, Alain DERYCKE : "**A distributed architecture for cooperative systems supporting public-access**", document interne au laboratoire Trigone, 1992.
- [VIEVI 93] C.VIEVILLE, A. KAYE, F. HOOGSTOEL : "**A Collaborative Learning Activity Server as the heart of a Communication System**", in Actes de Telepresence'93, "First International Conference in Technologies and Theories for Human Cooperation, Collaboration, Coordination", APPLICA'93, Lille, France, 22-24 Mars 1993, pp. 245-260.
- [VIEVI 94a] Claude VIEVILLE, Frédéric HOOGSTOEL, Pascal CROISY, Alain DERYCKE : "**A Distributed Architecture for Systems Supporting Public Access**", rapport interne laboratoire TRIGONE, 1994.
- [VIEVI 94b] Claude VIEVILLE : "**Modélisation des activités de collaboration synchrones et asynchrones**", in [GANYM 94].

[VILER 92] Pierre VILERS : "**Etude d'un système de téléconférence en temps réel : application à un jeu de rôle en gestion de production**", Thèse de doctorat soutenue à l'Université des Sciences et Technologie de Lille en Février 1992 (Laboratoire Trigone).

[WEGNE 92] Peter WEGNER : "**Dimensions of Object-Oriented Modeling**", in COMPUTER, IEEE, octobre 1992.

[WEI 92] Yi-hsiu WEI, Chuan-lin WU : "**Integrating RPC and Message Passing for Distributed Programming**", in Open Distributed Processing, J. De Meer, V. Heymer, R. Roth (Editors), Elsevier Science Publishers B.V. (North Holland), Actes de IFIP TC6/WF6.4 International Workshop on Open Distributed Processing, Berlin, Allemagne, 8-11 Octobre 1991.

[WINOG 86] T. WINOGRAD : "**A language/Action perspective on the design for cooperative works human-computer interaction**", Actes de CSCW'86, ACM, 1986.

[ZACHA 90] Wayne W. ZACHARY, Scoot P. ROBERTSON : "**Introduction to Cognition, Computation, and Cooperation**", in [ROBER 89], pp. 1-24.

LISTE DES ILLUSTRATIONS

Figure 1. Le modèle de l'activité de Engeström.....	10
Figure 2. Le trèfle SCOOP.....	13
Figure 3. La complexité d'évaluation d'un dispositif informatisé d'apprentissage coopératif [SMITH 94]	24
Figure 4. Profil des applications de la classe Timbuktu	30
Figure 5. Profil des applications de la classe Team WorkStation.....	31
Figure 6. Profil des applications de la classe ShX.....	32
Figure 7. Profil des applications de la classe ShowMe.....	33
Figure 8. Profil des systèmes de conférence multimédia.....	35
Figure 9. Profil des applications de la classe Information Lens	36
Figure 10. Profil des applications de la classe Andrew Multi-media Message System.....	37
Figure 11. Profil de SuperCOM.....	38
Figure 12. Profil de The Co-ordinator	40
Figure 13. Profil des applications de la classe CHAOS.....	41
Figure 14. Profil de Cooperative SEPIA.....	43
Figure 15. Profil d'Alliance.....	44
Figure 16. Une salle virtuelle de coopération MILAN [HAMMA 91].....	45
Figure 17. Profil des systèmes de la classe MILAN	47
Figure 18. Profil de The Collaborative Desktop.....	48
Figure 19. Profil de Pages.....	51
Figure 20. Profil de ConversationBuilder.....	52
Figure 21. Profil de ConversationBuilder amélioré.....	53
Figure 22. Récapitulation de la participation des collecticiels orientés interface aux enjeux du support de la collaboration	55
Figure 23. Participation des messageries aux enjeux du support de la collaboration.....	56
Figure 24. Participation des systèmes de conversation aux enjeux du support de la collaboration	57
Figure 25. Récapitulation de la contribution des systèmes orientés information au support de la collaboration.....	58
Figure 26. Synthèse de la participation des systèmes intégrés aux enjeux du support de la collaboration.....	59
Figure 27. Les trois niveaux de notification	70
Figure 28. Les filtres de Terry.....	85
Figure 29. Séparer et Relier les Espaces de Conversation et d'Information	87
Figure 30. La variété des situations d'apprentissage [SMITH 94].....	102
Figure 31. Les six points de vue de MOCCA [MOCCA 90]	111

Figure 32. Le modèle ORDIT de l'organisation [DOBSO 91]	113
Figure 33. Le modèle AMIGO, extrait de [DANIE 89].....	117
Figure 34. Articulation de MOCCA et ODP, traduit de [MOCCA 92].....	130
Figure 35. La pyramide de l'organisation.....	134
Figure 36. Supporter divers environnements de coopération [SMITH 94]	135
Figure 37. Modélisation de type ORDIT de l'organisation des situations d'apprentissage dans un institut d'enseignement ouvert.....	137
Figure 38. Les participants d'une réunion mènent de nombreuses activités [SMITH 94]	139
Figure 39. Caractérisation des espaces de coopération.....	141
Figure 40. La communication entre les entités de l'Activité de Partage de Ressource selon AMIGO	145
Figure 41. L'évolution d'une Conversation pour l'Action	175
Figure 42. Modélisation de type ORDIT de l'organisation des modules de formation d'un institut d'enseignement.....	181
Figure 43. Modélisation de type ORDIT de l'organisation générale d'un institut de formation.....	183
Figure 44. Interface dynamique du télépointeur	184
Figure 45. Interface de l'outil d'exploration de l'organisation.....	185
Figure 46. Interface de l'outil de gestion des actions	186
Figure 47. Interface de l'outil d'aide à la conversation.....	187
Figure 48. Interface de l'outil de gestion des rendez-vous du groupe.....	188
Figure 49. Modèle ontologique du domaine organisationnel	190
Figure 50. Le modèle ontologique de la coopération en temps différé.....	191
Figure 51. Le Modèle Ontologique de la Conversation.....	192
Figure 52. Le domaine de la documentation du modèle ontologique.....	193
Figure 53. Le domaine de l'information partagée du modèle ontologique	194
Figure 54. Le modèle de la Salle Virtuelle dans le Système d'Information de Co-Learn	198
Figure 55. Le domaine de la communication dans le Système d'Information de Co-Learn	201
Figure 56. Le plan d'un centre de ressources virtuelles Co-Learn	202
Figure 57. Le schéma générique du domaine de l'organisation du modèle d'information.....	203
Figure 58. Spécialisation des classes génériques du domaine de l'organisation.....	204
Figure 59. Le domaine de l'expertise utilisateur du modèle d'information.....	206
Figure 60. Les relations entre les outils médiatisées par les objets d'information	207
Figure 61. Architecture d'une Application Interactive Mono-Utilisateur	211
Figure 62. Architecture du Partage de la Vue d'une Application Interactive.....	212

Figure 63. Architecture d'une application interactive de groupe	214
Figure 64. Architecture d'un espace de coopération	217
Figure 65. Architecture Logicielle de Co-Learn	220
Figure 66a. Coopération des objets distribués Co-Learn lors de l'entrée d'un participant dans une Salle Virtuelle	222
Figure 66b. Les méthodes des objets Smalltalk activées lors de l'entrée d'un participant dans une Salle Virtuelle	225
Figure 67. Coopération des objets distribués du réseau Co-Learn lors d'une requête dans l'activité de conférence	226
Figure 68. L'architecture de communication sur les clients.....	231
Figure 69. Extrait du script Smalltalk d'installation d'une organisation Co-Learn ...	234
Figure 70. Extraits du code Smalltalk assurant la persistance sur le serveur	234
Figure 71. Les composants technologiques d'un serveur Co-Learn.....	237
Figure 72. Le réseau physique Co-Learn.....	239
Figure 73. Articulation des espaces dans une situation de production d'information partagée en mode séparé	248
Figure 74. Articulation des espaces dans une situation Co-Learn de réunion en temps réel	249

LISTE DES TABLEAUX

Table	Page
Table 1. Le support de la communication	17
Table 2. Répartition des fonctionnalités parmi les enjeux.....	26
Table 3. Récapitulation de l'évaluation.....	54
Table 4. Mécanismes avancés de bases de données et modes de coopération.....	77
Table 5. Les points de vue sur l'organisation.....	122
Table 6. Les points de vue sur l'information.....	125
Table 7. Les points de vue dynamiques	127
Table 8. Les points de l'ingénierie	129
Table 9. Les points de vue de la technologie	129
Table 10. Exemple de spécialisations de Salles Virtuelles.....	142
Table 11. Messages de Conférence et Messages de Conversation	163

