
1. Introduction

Le domaine du Travail Coopératif Assisté par Ordinateur (TCAO) a pour thème d'étude les systèmes interactifs multi-utilisateurs permettant à plusieurs utilisateurs de travailler ensemble. Les termes **collecticiel** et **synergiciel** désignent ce type de système interactif. La terminologie anglo-saxonne emploie les termes de *groupware* pour désigner un collecticiel et de *Computer Supported Cooperative Work (CSCW)* pour désigner le domaine du TCAO.

De nombreuses définitions ont été proposées pour caractériser un collecticiel dont nous citons la plus courante, celle de C. Ellis :

“Computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment.” [Ellis 1991]

“Les collecticiels sont des systèmes informatiques qui assistent un groupe de personnes engagées dans une tâche commune (ou but commun) et qui fournissent une interface à un environnement partagé.” (traduction de A. Karsenty [Karsenty 1994])

Souvent perçu comme la "petite sœur" de l'IHM, le domaine de la TCAO doit adapter et étendre les méthodes et outils développés en IHM car ceux-ci ont été forgés par rapport à une activité mono-utilisateur et en fonction d'un seul utilisateur. Ce constat est particulièrement saillant en ce qui concerne les facteurs humains (sciences humaines). Tandis que le domaine de l'IHM s'est longtemps appuyé sur des théories de psychologie cognitive et comportementale, le domaine du TCAO dès ses débuts s'est tourné vers les sciences sociales et en particulier l'ethnographie pour

analyser les aspects sociaux au sein d'un groupe en situation de travail. Nous constatons encore aujourd'hui un gouffre entre les études liées aux facteurs humains en IHM et en TCAO. Dans [Bannon 1998], L. Bannon souligne cette différence et prône pour une plus grande socialité des études en IHM en prônant une "ergonomie sociale".

Dans cette étude, nous nous intéressons plus particulièrement aux outils conceptuels et aux outils de développement pour la mise en œuvre logicielle des collecticiels. Dans le domaine du TCAO, ces outils ont une position essentielle du fait de la complexité de mise en œuvre des collecticiels. Comme nous l'avons évoqué dans l'introduction générale, cette complexité est accrue comparé au développement d'un système interactif mono-utilisateur en IHM classique. D'une part, le collecticiel s'adresse à un groupe d'utilisateurs : celui-ci doit être capable de reproduire un espace social et un climat social favorable à l'activité de groupe. A cela viennent se greffer des problèmes d'ordre éthique comme le respect de la personne et de sa vie privée. D'autre part, le développeur d'un collecticiel doit faire face à des difficultés techniques supplémentaires liées à la gestion des accès concurrents et aux problèmes induits par la communication à travers un réseau informatique.

L'objectif de ce chapitre est de montrer le rôle et l'apport des outils de conception et de développement logiciel des collecticiels pour répondre à cette complexité et montrer dans quelle mesure ces outils aident le concepteur puis le développeur dans la réalisation d'un collecticiel. Ainsi, pour positionner ces outils dans un contexte de réalisation d'un collecticiel, ce chapitre est organisé de la façon suivante : la première partie présente trois taxonomies pour caractériser les collecticiels selon trois points de vue et ainsi mesurer la complexité de mise en œuvre d'un collecticiel. Dans la seconde partie du chapitre, nous présentons un ensemble de méthodes et outils dédiés à la conception et à la réalisation de collecticiels. A des fins analytiques et pour souligner leurs portées, nous les situons au sein du cycle de vie du logiciel. Enfin, la dernière partie rappelle, à la lumière de ce chapitre, les objectifs de cette étude et détaille la démarche de travail retenue pour traiter notre sujet.

2. Taxonomies des collecticiels

Dans cette partie, nous présentons trois taxonomies aux caractéristiques complémentaires afin de définir plus finement ce qu'est un collecticiel et cerner l'étendue des possibilités. La multiplicité des possibilités peut se

voir comme un indicateur du dynamisme de ce domaine, avec, sa compagne immédiate, la complexité de réalisation logicielle.

Les deux premières taxonomies présentées, l'une selon les types d'application et l'autre Espace-Temps, sont incontournables dans la littérature du domaine. La dernière taxonomie choisie repose quant à elle sur un modèle du travail coopératif. Les trois taxonomies exposées, notre première contribution sera de choisir et proposer un ensemble de termes issus de ces taxonomies qui couvrent raisonnablement le domaine de réflexion.

2.1. TYPES D'APPLICATION

La première taxonomie consiste en une classification par domaines d'application. Aussi cette taxonomie doit souvent être révisée pour prendre en compte les nouveaux types d'application. Elle traduit donc une image du domaine à un instant donné. Pour élaborer une liste des domaines d'application des collecticiels, nous avons combiné plusieurs taxonomies existantes [Ellis 1991] [Dix 1993] [Karsenty 1994] [UsabilityFirst 2002]. La variété des domaines d'application souligne le dynamisme de cet axe de recherche. Nous obtenons quatre catégories de collecticiels qui sont :

- 1 les applications dédiées à la communication homme-homme médiatisée (CHHM, ou *CMC* pour *Computer-Mediated Communication*) où nous regroupons les messageries électroniques, les forums de discussion, les systèmes de vidéoconférence et les *mediaspace*,
- 2 les applications d'édition où nous classons les éditeurs de texte et les tableaux blancs partagés,
- 3 les applications pour la coordination où nous rassemblons les systèmes *workflow*, les systèmes d'aide à la décision et les calendriers partagés,
- 4 les applications de jeux en réseau.

Nous présentons dans la suite les différents types de collecticiel dans l'ordre des quatre catégories ci-dessus.

- **Les messageries électroniques** (*email*) sont actuellement les collecticiels les plus répandus et les plus utilisés. Le trafic généré par ces systèmes est devenu tel que, notamment avec l'intrusion du courrier électronique publicitaire (*junk mail* ou *spam*), les outils de messagerie se sont enrichis de fonctionnalités "intelligentes" pour trier les courriers, pour détruire les courriers non désirables ou pour envoyer des réponses automatiquement. Il existe de nombreuses applications de gestion de courriers électroniques tels que l'utilitaire *mail* sous Unix ou l'application Eudora [Qualcomm 2002]. La grande tendance dans

ce domaine est celle de la messagerie instantanée (*Instant Messaging*) avec, par exemple, les services de messagerie SMS (*Short Messaging Services*) dans le monde de la téléphonie mobile.

- **Les forums de discussion** (*chat* et *newsgroup*) : les deux principales classes de forum de discussion diffèrent par leur mode d'utilisation, synchrone ou asynchrone. La première classe regroupe les forums en ligne du type IRC (*Internet Relay Chat*), des applications très répandues qui reposent sur le concept de canal à thème (canal de discussion). La discussion est ici synchrone à l'opposé de la seconde classe qui rassemble les listes de diffusion (*mailing list*) et les *newsgroups* (système *USENET*) pour des discussions asynchrones sur un thème donné.
- **Les systèmes de vidéoconférence** permettent à des personnes physiquement distantes de se réunir et communiquer par l'intermédiaire d'un support audio et vidéo. Il s'agit d'un forum de discussion offrant une communication reposant sur des données audio et vidéo à l'opposé des forums de discussion du point précédent qui se basent sur des échanges textuels. La grande difficulté du déploiement de ce type d'application est liée en grande partie à la nécessité de disposer d'une bande passante capable de diffuser et recevoir des données audio et vidéo avec une qualité acceptable. Le système NetMeeting est un des systèmes grand public le plus connu [Microsoft 2002].

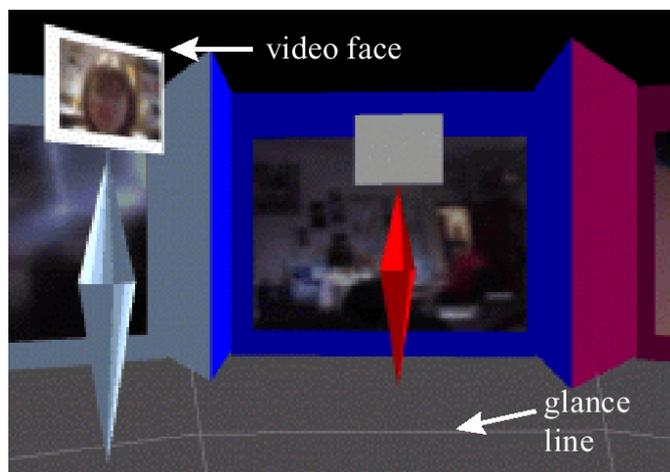


Figure 1

Système de vidéoconférence CVE [Reynard 1998]

Ce système immerge des utilisateurs dans un espace en trois dimensions. Les utilisateurs sont observables à travers une vignette vidéo collée à l'extrémité d'une forme géométrique longiligne. Cette forme, avatar correspondant à un utilisateur, peut être déplacée dans l'espace par l'utilisateur. Il est aussi possible d'observer les différents bureaux à travers des "fenêtres" vidéo.

- **Les mediaspace** [Coutaz 1999] [Dourish 1992] sont des collecticiels mettant en œuvre une liaison vidéo au sein d'une équipe dans le but de favoriser la communication informelle et d'entretenir une conscience de groupe forte entre membres distants : entre deux étages, deux bâtiments, deux villes, etc. L'objectif visé est différent des systèmes de vidéoconférence bien que les deux types d'application reposent sur des flux vidéo. En effet, contrairement à la vidéoconférence qui met en

relation des individus sur une courte période et de manière planifiée, la connexion vidéo d'un *mediaspace* est permanente et l'interaction est opportuniste. Aussi un *mediaspace* soulève le problème de la protection de la vie privée de par la présence de caméras fonctionnant en permanence dans les bureaux ou lieux communs d'un organisme (cafétéria, salle de réunion, etc.).

Notons enfin que certains systèmes comme le système CVE [Reynard 1998] de la Figure 1, combinent à la fois un système de vidéoconférence et un *mediaspace*.

- **L'édition conjointe** (*shared editing*): les éditeurs partagés sont des systèmes dédiés à l'édition collaborative de documents avec gestion des différentes versions. Ces outils sont complexes à réaliser, en particulier pour la gestion des tâches concurrentes comme le "défaire" et "refaire" (*undo* et *redo*) [Choudary 1995] ou la fusion de différentes versions [Dourish 1996b]. Les éditeurs de texte partagés comme le système StorySpace [Eastgate 2002] ou les éditeurs de dessins partagés comme les tableaux blancs partagés [Ishii 1994] (*shared whiteboard*) sont des exemples de collecticiels permettant l'édition conjointe.
- **Les systèmes workflow** [Bowers 1995] sont des systèmes dédiés à la gestion de processus industriels et à la coordination des différents intervenants au cours d'un processus. Un processus s'articule sur la réalisation de documents industriels et le système *workflow* a la charge de veiller à la bonne circulation des documents entre les différents intervenants aux moments clés du processus.
- **Les calendriers partagés** (*group calendars*) sont des systèmes qui offrent des services de planification de tâches, de gestion de projets et de coordination de membres d'une équipe de travail. Contrairement aux systèmes *workflow*, la planification n'est pas centrée sur l'acheminement d'un document ou d'un quelconque support de travail. Les fonctionnalités usuelles incluent la détection d'incompatibilités dans la planification d'une tâche ou la détermination de plages horaires communes aux membres d'un groupe. Par exemple, Lotus/Organizer [Lotus 2002] est un outil de planification collaborative.
- **Les systèmes d'aide à la décision** (*GDSS, Group Decision Support Systems*) sont conçus pour faciliter la prise de décisions grâce à l'apport de nombreux outils: brainstorming, votes, pondération des décisions, génération et annotation des idées, etc. Ces systèmes encouragent tous les participants à s'engager dans la prise de décision, par exemple en permettant de conserver l'anonymat ou en garantissant



Figure 2
Starcraft [Blizzard 2002]

Starcraft est un exemple de jeu de stratégie fondé sur la compétition. Les joueurs doivent élaborer une tactique de développement de troupes pour détruire l'adversaire et pour se protéger de l'invasion ennemie.

que chaque participant puisse prendre au moins une fois la parole [UsabilityFirst 2002].

- **Les jeux en réseau** sont certainement les collecticiels qui connaissent, avec les systèmes de messagerie, l'essor le plus fulgurant. Parmi les jeux les plus connus, citons le jeu Quake [Id Software 2002] ou Starcraft [Blizzard 2002] présenté à la Figure 2. Ces jeux misent sur la coopération et la compétition entre les joueurs. Ce type d'application est comparable à une forme d'éditeur partagé reposant, dans la majorité des cas, sur un mode de communication textuelle.

En synthèse, nous constatons une grande variété des domaines d'application des collecticiels. Comme expliqué en introduction de ce paragraphe, nous proposons de regrouper les neuf domaines d'application ci-dessus présentés, en quatre catégories : communication (messagerie, forum, vidéoconférence, *mediaspace*), édition (éditeur partagé, tableau blanc partagé), coordination (*workflow*, calendrier partagé, prise de décision) et jeux. Certes cette catégorisation traduit l'ensemble des collecticiels existants à ce jour. A ne pas en douter, il conviendra de revisiter notre taxonomie dans quelques années pour intégrer de nouveaux types de collecticiels.

De plus, notons que certains collecticiels peuvent couvrir plusieurs domaines d'application au sein d'une catégorie. Par exemple, le système CVE de la Figure 1 est à la fois un système de vidéoconférence et un *mediaspace* au sein de la catégorie Communication, puisque le système CVE est dédié à la communication homme-homme médiatisée.

Relevons enfin le cas de collecticiels qui embrassent plusieurs domaines d'application de plusieurs catégories. L'objectif affiché de tels collecticiels est alors l'intégration de plusieurs outils complémentaires pour collaborer

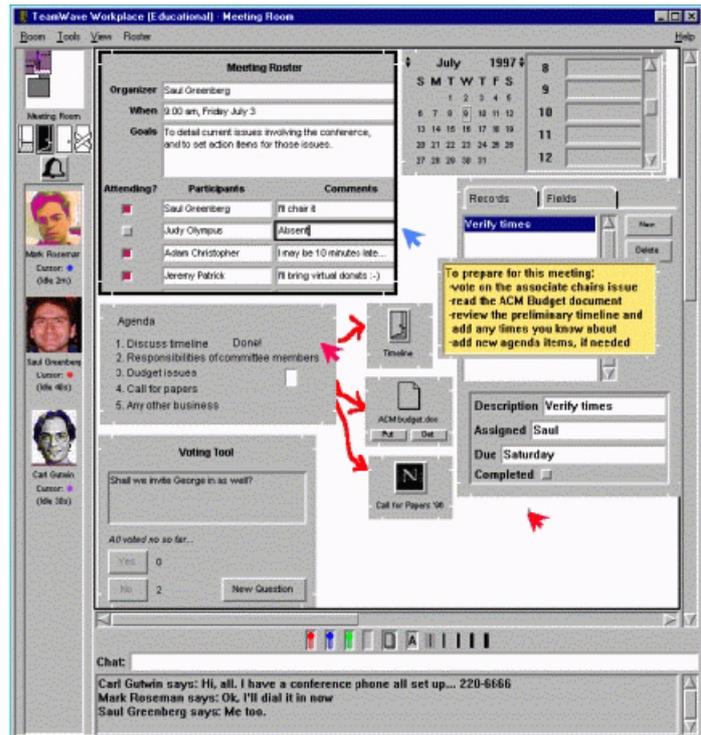


Figure 3
Bureau intégré TeamRooms [Roseman 1996b]

Il s'agit d'un bureau complet mettant à disposition un ensemble d'outils tels que les annotations, le système de vote, un tableau blanc, un calendrier partagé, une fenêtre de discussion, etc. Ce système rend observable les activités de tous les participants par le biais de cartes d'identité (photos), de pointeurs ou de vue radar (en haut à gauche de la copie d'écran).

comme des éditeurs partagés, des tableaux blancs partagés et des systèmes d'aide à la décision au sein d'un même environnement. Ce dernier peut être réel ou virtuel.

- Dans le premier cas (environnement d'intégration réel), il peut s'agir d'une pièce spécialement aménagée pour faciliter le face-à-face et l'interaction avec les documents électroniques comme, la *Facilitator Room* du MIT MediaLab [MIT Medialab 2002].
- Dans le second cas, où les outils de collaboration sont intégrés dans un environnement virtuel, le système s'appuie sur la métaphore de la pièce ou du bureau partagé, comme le système TeamRooms [Roseman 1996b] de la Figure 3. Grâce à un espace partagé entre les utilisateurs, le système rend perceptible la présence de tous les utilisateurs, les activités de chacun, l'historique des activités et les communications entre utilisateurs.

Par l'exposé de cette taxonomie, nous nous sommes intéressés aux domaines d'application d'un collecticiel. Nous considérons maintenant deux autres caractéristiques complémentaires pour caractériser un collecticiel : les aspects temporel et spatial de la collaboration.

2.2. CLASSIFICATION ESPACE-TEMPS

La classification Espace-Temps repose sur deux caractéristiques, à savoir où et quand une action est exécutée par un des utilisateurs par rapport aux autres utilisateurs. Il s'agit de la classification la plus largement adoptée dans le domaine du TCAO, nommée Espace-Temps ou matrice Espace-

	<i>Espace</i> ↑			
Lieux différents (imprévisible)	3	6	9	
	Rencontre informelle	Edition partagée	Newsgroup Mél (accès WWW) SMS	
Lieux différents (prévisible)	2	5	8	
	Vidéoconférence	Workflow	Mél	
Même lieu	1	4	7	
	Jeu sur console	Relais, enchaînement tour de parole	Post-It	
	← Même moment	Moments différents (prévisible)	Moments différents (imprévisible)	→ <i>Temps</i>

Figure 4
 Classification Espace-Temps

Temps. La classification présentée à la Figure 4 est celle de J. Grudin [Grudin 1994], une version étendue de la matrice Espace-Temps de C. Ellis [Ellis 1991].

La classification Espace-Temps s'organise selon deux axes caractérisant l'usage du collecticiel : le premier axe *Espace* considère la distance spatiale entre les utilisateurs (*Même lieu* et *Lieux différents*) et le deuxième axe *Temps* considère la distance temporelle entre les utilisateurs (*Même moment* et *Moments différents*). Pour ce dernier axe, les termes couramment utilisés sont **synchrone** et **asynchrone**. Néanmoins, il n'est pas toujours possible d'attester du caractère prévisible du moment d'interaction, ni du lieu dans lequel se déroule l'interaction puisque l'interaction peut se dérouler au cours du temps dans des lieux différents. Ainsi, l'extension de la matrice Espace-Temps [Ellis 1991] proposée par J. Grudin [Grudin 1994] tient compte du caractère imprévisible de l'interaction pour les deux axes considérés. La matrice résultante, comme le montre la Figure 4, considère trois cas pour chaque axe. Nous illustrons les neuf types de collecticiels obtenus :

- 1 Les consoles de jeux ou les bornes d'arcade multi-joueurs sont des formes de collecticiel synchrone en face-à-face. Ces consoles sont constituées de deux manettes (ou plus) et chaque joueur pilote un personnage de jeu ou un objet animé (voiture, raquette, etc).
- 2 Les systèmes de vidéoconférence sont des collecticiels synchrones et l'interaction entre les utilisateurs a lieu en temps réel (même moment). Les participants sont répartis géographiquement dans des lieux différents mais prévisibles.
- 3 La rencontre informelle, comme celle au sein d'un forum de discussion, se réalise de manière imprévisible comme une rencontre

fortuite dans la rue. Ce type d'interaction est de nature synchrone. Par exemple, les *mediaspace* [Coutaz 1999] [Dourish 1992] ont pour but de favoriser les rencontres informelles.

- 4 La prise de parole au cours d'une présentation est un cas où l'enchaînement des actions a été planifié. Il s'agit en l'occurrence de prendre le relais les uns à la suite des autres pour continuer la présentation. Contrairement aux systèmes *workflow* qui relèvent de la situation suivante, les actions réalisées se déroulent dans un même lieu.
- 5 Les systèmes *workflow* ont pour objectif la coordination des activités et des intervenants au cours d'un processus industriel. Ces systèmes permettent ainsi la planification des interventions de chacun sur les documents industriels échangés au cours d'un processus. Dans cette situation, les participants sont répartis géographiquement mais localisables (le bureau sur le lieu de travail).
- 6 Dans le cas de l'édition partagée asynchrone, il n'est pas nécessaire de connaître la localisation exacte des différents auteurs du document. Par contre, pour pouvoir transmettre le document entre les auteurs, ces derniers doivent planifier leurs échanges.
- 7 Le post-it, qu'il soit réel (papier) ou virtuel, est un support de la collaboration très largement adopté. La transmission du post-it est totalement imprévisible et se déroule, au sens large, dans un même lieu. Par extension, nous pouvons supposer que le support d'un post-it, comme un livre, est toujours le lieu de l'interaction bien qu'il soit possible de le déplacer.
- 8 Le courrier électronique dans sa forme usuelle est comparable au courrier postal : les destinataires sont répartis géographiquement, mais sont localisables (lieu de travail, lieu de résidence, etc). Ainsi, lorsque l'on envoie un message électronique, l'expéditeur peut localiser le destinataire. Par contre, le moment de réception est imprévisible.
- 9 Les listes de diffusion (ainsi que les nouvelles formes de courrier électronique avec accès par le WWW) ne permettent pas la localisation des participants. De plus, il n'est pas possible de prévoir à quel moment les messages seront lus.

En synthèse, cette classification considère deux caractéristiques, le temps et l'espace, pour définir 9 classes de collecticiels, que nous avons illustrées.

Il convient de noter que cette classification Espace-Temps ne permet pas de différencier les collecticiels où les utilisateurs sont mobiles. En effet, cette classification ne tient pas compte du passage d'un lieu à un autre de manière continue, ce qui est désormais possible avec l'informatique mobile.

Enfin il est intéressant de constater que les collecticiels existants classés par domaine d'application dans le paragraphe précédent considèrent presque tous que leurs utilisateurs sont distants (lieux différents, cas prévisible et imprévisible). Certes, il semble assez immédiat de conclure que c'est dans le cas d'un groupe d'utilisateurs distants que l'informatique peut le plus contribuer à favoriser la collaboration. Il convient néanmoins de noter aussi que la réalisation de collecticiels, où les utilisateurs sont distants, soulève des enjeux techniques que la communauté de recherche a souhaité traiter. Nous pouvons faire la comparaison entre le domaine de la TCAO et l'interaction multimodale. En effet, en multimodalité, nombreuses sont les recherches sur l'usage synergique des modalités [Oviatt 1999] car il s'agit du cas sans doute le plus difficile à réaliser et non du cas le plus utile. De manière similaire, nombreux sont les travaux en TCAO sur les collecticiels où les utilisateurs sont distants : ce constat n'implique pas que les collecticiels pour des utilisateurs regroupés en un seul lieu ne soient pas utiles.

2.3. TAXONOMIE DU TRAVAIL COOPÉRATIF

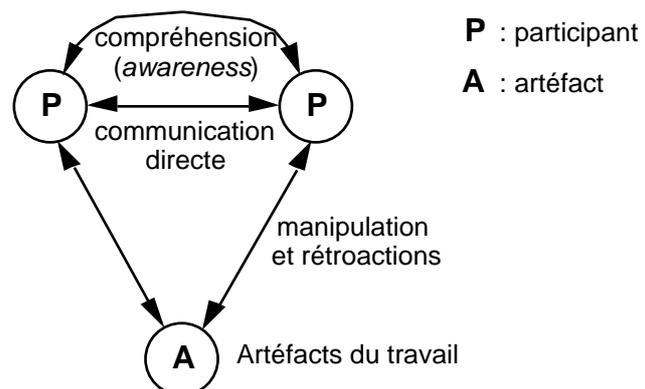


Figure 5
Taxonomie du travail coopératif [Dix 1993]

La taxonomie des collecticiels selon A. Dix [Dix 1993] repose sur un modèle du travail coopératif. Ce modèle, comme le montre la Figure 5, identifie deux entités impliquées dans le travail coopératif : les participants (P) et les artéfacts du travail (A), c'est-à-dire les entités manipulées permettant aux participants d'interagir entre eux et avec le système. Par exemple, ces artéfacts peuvent être les outils mis à disposition comme un outil pinceau utilisé pour dessiner dans un tableau partagé.

Ce modèle identifie les relations entre ces entités, c'est-à-dire les participants et les artéfacts du travail, qui caractérisent le travail coopératif. Il existe trois types de relation :

- la communication directe entre les participants au cours d'une activité (orale, écrite, gestuelle, etc),

- l'établissement d'une compréhension mutuelle pour mener des actions conjointes, et,
- la manipulation des artefacts.

Le but de la communication directe est d'échanger des informations mais aussi d'établir une compréhension mutuelle des actions qui vont être menées conjointement. Cette notion de compréhension mutuelle est essentielle pour le succès d'une action conjointe. La manipulation d'artefacts évoque la manipulation d'objets physiques que l'on déplace ou modifie sur un bureau réel comme déplacer un livre, écrire sur une feuille, utiliser son téléphone, etc. La rétroaction est la perception par les autres utilisateurs du résultat de ces actions. Par exemple, s'apercevoir qu'un livre a été déplacé ou que quelque chose a été écrit sur la feuille.

A partir de ce modèle, A. Dix a identifié trois catégories [Dix 1993] de collecticiels (taxonomie du travail coopératif) relatives aux trois relations mises en évidence par ce modèle du travail coopératif :

- **Les systèmes de communication Homme-Homme médiatisée (CHHM)** désignent les systèmes dédiés à la communication directe entre les participants. Cette catégorie de collecticiels regroupe, entre autre, le courrier électronique, les forums de discussion, la vidéoconférence et les *mediaspace*. Nous retrouvons ici notre catégorie de la première taxonomie.
- **Les systèmes de réunions et d'aide à la décision** sont des collecticiels dont le but est de favoriser et d'aider à la compréhension mutuelle pour faciliter le déroulement de réunions ou la prise de décisions entre différents participants. Cette catégorie regroupe donc les systèmes d'aide à la décision (GDSS) et les systèmes de réunion virtuelle (bureau partagé) et réelle (tableau blanc réel et partagé).
- **Les systèmes d'espaces partagés** (*shared workspaces*) recouvrent les collecticiels mettant en œuvre des espaces partagés dans lesquels les participants peuvent manipuler des artefacts. Il s'agit par exemple des éditeurs partagés ou des calendriers partagés.

En conclusion, les trois classes identifiées par cette taxonomie, basée sur un modèle du travail coopératif, pour caractériser un collecticiel (communication directe, compréhension mutuelle et manipulation d'artefacts partagés) sont complémentaires aux dimensions espace et temps de la taxonomie précédente. En effet, considérons la communication directe : cette dernière peut concerner les neuf catégories de la taxonomie Espace-Temps, que ce soit en face-à-face (cas 1), par le biais de post-it (cas 7) ou par les notes prises dans le cas d'une édition

partagée (cas 6). De plus, nous constatons que les neuf domaines d'application de la première taxonomie peuvent être rangés dans les trois classes identifiées ici.

2.4. SYNTHÈSE

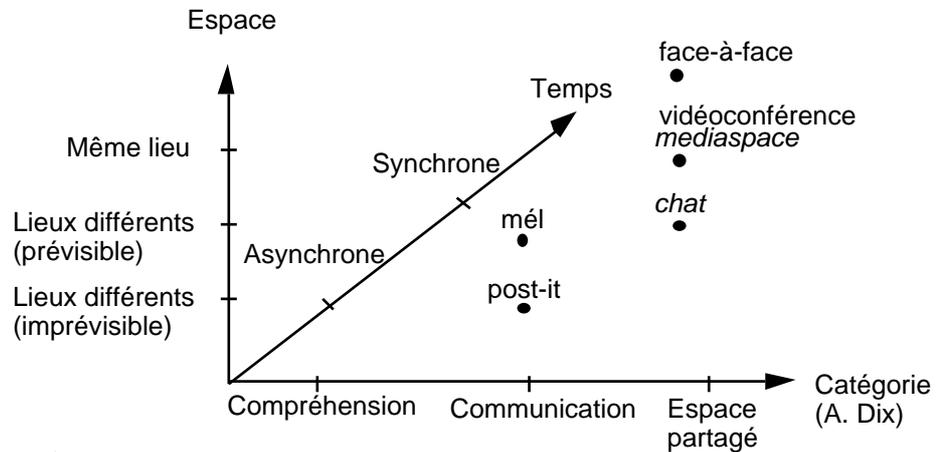


Figure 6
Schéma de synthèse des taxonomies

En synthèse, nous avons présenté trois taxonomies pour souligner l'étendue des possibilités dans le domaine du TCAO. Au cours de leur présentation successive, nous avons souligné leur complémentarité. A la Figure 6 de synthèse, nous combinons les axes de classification des deux dernières taxonomies. En effet nous partons des neuf cas de la taxonomie Espace-Temps et nous affinons chacune des cases de la matrice par les trois types de la taxonomie de A. Dix. La première taxonomie, reposant sur les domaines d'application, est par définition instable et doit être enrichie au fur et à mesure qu'un nouveau collecticiel traite d'un domaine d'application non encore répertorié. De plus, nous pouvons situer les neuf domaines d'application cités au sein de ces trois axes. A titre illustratif, nous situons à la Figure 6, les types de collecticiels de la première taxonomie que nous avons rangés sous la catégorie Communication dans le paragraphe 2.1.

En étudiant les taxonomies existantes des collecticiels, nous avons souligné leur diversité en terme de domaines d'application et d'usage. Cette diversité laisse entrevoir la difficulté de mise au point de méthodes et d'outils pour la conception et la réalisation de tels systèmes. Méthodes et outils pour la mise en œuvre des collecticiels font l'objet de la partie suivante.

3. Outils pour la mise en œuvre d'un collectifiel

Notre étude porte sur une catégorie d'outils, les outils de conception et de réalisation logicielles des collectifiels. Aussi dans les chapitres suivants, nous dressons un état de l'art ainsi qu'une analyse critique des outils existants de cette catégorie. Dans cette partie, nous souhaitons définir cette catégorie d'outils en cernant sa portée par rapport aux autres outils existants. Pour cela nous présentons un ensemble de méthodes et d'outils au sein d'un cadre fédérateur. L'objectif n'est donc pas ici de dresser un état de l'art mais de situer des outils existants les uns par rapport aux autres.

A des fins analytiques, nous avons choisi de présenter les méthodes et outils existants selon le cycle de vie du logiciel en V. Nous ne prenons pas position sur le fait que ce cycle est adapté ou non à la mise en œuvre d'un collectifiel. Nous nous servons de ce cycle de vie comme un canevas structurant. En particulier ce cycle de vie établit une distinction nette entre la conception ergonomique d'un collectifiel, l'espace IHM, et la conception et réalisation logicielles, l'espace logiciel, notre sujet d'étude.

Le paragraphe suivant présente les principales étapes du cycle en V et définit ainsi l'organisation de cette partie.

3.1. CYCLE DE VIE LOGICIEL

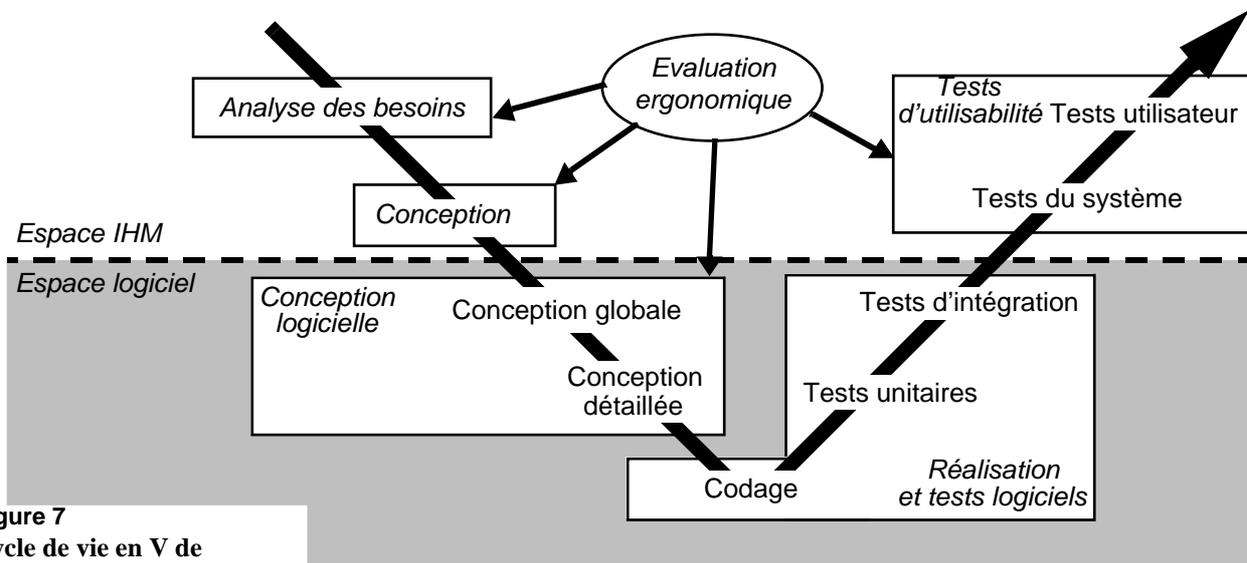


Figure 7
Cycle de vie en V de développement logiciel

Les étapes du cycle de vie en V [McDermid 1984] sont représentées à la Figure 7. Nous avons adapté le cycle de vie au développement logiciel de systèmes interactifs. Il existe d'autres cycles de vie comme le cycle en cascade [Royce 1970] ou en spirale [Boehm 1981]. Notre propos n'est pas de comparer les différents cycles de vie mais d'en extraire les étapes

principales afin de définir un canevas fédérateur pour situer ensuite les méthodes et outils dédiés aux collecticiels.

Les étapes de ce cycle peuvent être organisées selon deux catégories : les étapes relevant de l'espace IHM, c'est-à-dire les étapes de conception ergonomique du système et d'élaboration d'un modèle d'interaction répondant à des requis identifiés à l'issue de l'étape d'analyse des besoins, et les étapes relevant de l'espace logiciel, c'est-à-dire les étapes de conception et de développement logiciels. L'évaluation ergonomique, comme le montre la Figure 7, intervient à toutes les étapes du cycle. Le but de cette phase est d'identifier les incohérences et les erreurs d'ergonomie pour pouvoir les corriger le plus tôt possible.

L'objectif de l'étape d'analyse des besoins est de déterminer les besoins afin de produire un logiciel qui réponde aux attentes des utilisateurs [Assali 1998]. Les résultats de cette étape sont consignés dans un cahier des charges. Cette étape consiste donc à cerner le domaine d'application et le rôle attendu du système tout en prenant en compte les contraintes de développement et d'utilisation. Le paragraphe 3.2 présente cette étape à travers trois outils pour l'analyse des besoins : l'étude ethnographique, le modèle Denver et la méthode GTA pour l'analyse de la tâche.

L'étape suivante est alors la conception du système. C'est au cours de cette étape que l'on identifie les requis fonctionnels (spécifications fonctionnelles) et les requis utilisateurs (spécifications externes). Le document de spécifications externes issu de cette étape reflète le système tel qu'il sera utilisé et perçu par l'utilisateur. Le paragraphe 3.3 traite d'abord des outils pour les spécifications fonctionnelles en présentant un modèle, le modèle du trèfle pour identifier les fonctionnalités d'un collecticiel. Les spécifications externes sont ensuite abordées par l'exposé d'un ensemble de propriétés ergonomiques spécifiques aux collecticiels.

L'étape de conception logicielle est une étape préliminaire à la phase de codage. A l'aide d'outils pour la conception logicielle, présentés au paragraphe 3.4, cette étape consiste à construire l'architecture logicielle décrivant la structure du logiciel à développer ainsi que son comportement à l'exécution.

L'étape suivante de codage dont la finalité est de produire un logiciel exécutable fait l'objet du paragraphe 3.5. Nous y présentons les outils de développement.

Enfin, le paragraphe 3.6 traite des outils pour l'évaluation ergonomique. Cette phase intervient à toutes les étapes du cycle et a pour but d'identifier et de corriger le plus tôt possible les erreurs de conception pouvant nuire à l'utilisabilité et à l'utilité du futur système.

3.2. OUTILS POUR L'ANALYSE DES BESOINS

Au cours de la phase d'analyse des besoins, le concepteur identifie les concepts du domaine d'application et élabore un modèle de tâches décrivant l'interaction avec le futur système. Pour déterminer ces concepts et analyser la tâche, le concepteur dispose d'outils hérités, en grande partie, des sciences sociales et en particulier de l'ethnographie.

Notamment, dans la première section, intitulée "Etude ethnographique", nous traitons de l'étude ethnographique qui consiste à étudier le comportement et les pratiques d'un groupe d'individus, par exemple, à partir d'observations sur le terrain ou à partir de questionnaires.

Dans la seconde section, intitulée "Modèle Denver", nous présentons le modèle Denver en tant que moyen complémentaire à l'étude ethnographique pour caractériser les besoins à partir des résultats issus de l'étude. En effet, bien que ce modèle ne soit pas complètement abouti, celui-ci offre une base de travail intéressante pour l'analyse des besoins. Il définit et organise un ensemble de critères permettant l'analyse du travail collaboratif ; en particulier, il offre des axes pertinents pour définir les situations et protocoles d'interaction

Enfin, dans la troisième section, intitulée "GTA et analyse de la tâche", nous présentons la méthode GTA qui est une méthode d'analyse et de description de la tâche qui exploite les résultats d'une étude ethnographique afin d'élaborer un arbre de tâches.

Etude ethnographique

Les travaux entrepris dans le domaine du TCAO marquent une rupture avec les méthodes en IHM largement influencées par la psychologie cognitive et du comportement et se placent dans une optique résolument sociale de par l'influence de disciplines telles que la sociologie ou l'ethnographie. Ces dernières ont une place importante dans les travaux sur les systèmes collaboratifs. Selon L. Bannon [Bannon 1998], le défaut principal de l'approche cognitive liée aux facteurs humains en IHM est de considérer l'utilisateur isolément et de modéliser son comportement indépendamment du contexte d'utilisation, de sa façon de travailler. Les travaux issus du monde du TCAO tendent donc à remettre en cause l'influence majeure de la psychologie cognitive en IHM et prônent pour une approche plus "sociale" qui prendrait en compte le contexte et les aspects sociaux de l'interaction.

Plus précisément, selon son origine étymologique, l'ethnographie est une branche de la sociologie qui consiste à recueillir une description du fonctionnement d'un groupe d'individus dans leur environnement par rapport à leurs usages, leurs relations, leur histoire, leur culture, etc. Une étude ethnographique permet ainsi de comprendre les pratiques et les activités. Le travail principal de l'ethnographe consiste d'abord à analyser l'activité dont il doit noter tous les phénomènes les plus significatifs, puis à les analyser. La deuxième phase consiste alors à collecter les

informations sur les activités de chaque individu du groupe. L'étude ethnographique se doit de décrire : le cadre de l'interaction (la localisation, le statut, etc.), les règles régissant l'organisation du groupe (rendre compte de la hiérarchie, de la répartition des rôles, des protocoles de communication, etc.), les rencontres informelles, les événements inattendus qui modifient la nature de l'interaction au sein de ce groupe [Van Der Veer 1996]. Enfin, selon Anderson [Anderson 1994], "l'étude ethnographique est une forme de reportage bien plus qu'une collecte de données" et l'ethnographe interprète ce qu'il observe. Il s'agit de capter l'aspect social du travail de groupe à travers ses pratiques, sa culture, son histoire, etc. L'étude ethnographique est donc un support essentiel pour la conception d'un collectif car elle permet, par l'étude des tâches accomplies, du comportement des participants à la tâche, et de l'environnement, de comprendre le fonctionnement complexe de l'action réalisée. Contrairement à l'étude d'un seul utilisateur, l'étude du fonctionnement d'un groupe est plus complexe car elle doit intégrer le comportement de chaque utilisateur ainsi que l'interaction entre utilisateurs. Cela est d'autant plus difficile lorsque le nombre d'utilisateurs varie au cours du temps et qu'il n'y a, a priori, aucune limitation de ce nombre.

Modèle Denver

Le modèle Denver (*Denver model*) [Salvador 1996], résultat d'un groupe de travail (*workshop*) à la conférence *ACM CHI (Computer Human-Interaction)* en 1995, est une ébauche de méthode d'analyse des besoins pour les applications où la notion de groupe est un facteur dominant.

Ce modèle offre un cadre pour identifier la situation d'interaction et le protocole social d'interaction. Les résultats de l'étude ethnographique peuvent être transcrits en terme de situation d'interaction et de protocole d'interaction. Cette transcription a l'avantage d'organiser les résultats et de caractériser les différentes formes de situation et protocoles d'interaction au sein d'un groupe. Ce modèle est bien utilisé pour la phase d'analyse et aide le concepteur pour décrire les besoins.

La situation d'interaction et le protocole social d'interaction sont définis de la façon suivante :

- **Situation d'interaction** (*Interactive situation*)

Les situations d'interactions sont définies par les relations entre le temps, l'espace et le couplage de l'interaction (dépendance entre les participants). Les participants peuvent définir un petit ou un grand groupe, être proches ou loin, avoir des interactions spontanées ou au contraire prévues, être dépendants d'un autre participant dans la progression de leurs travaux ou ne pas l'être. Enfin ils peuvent être en

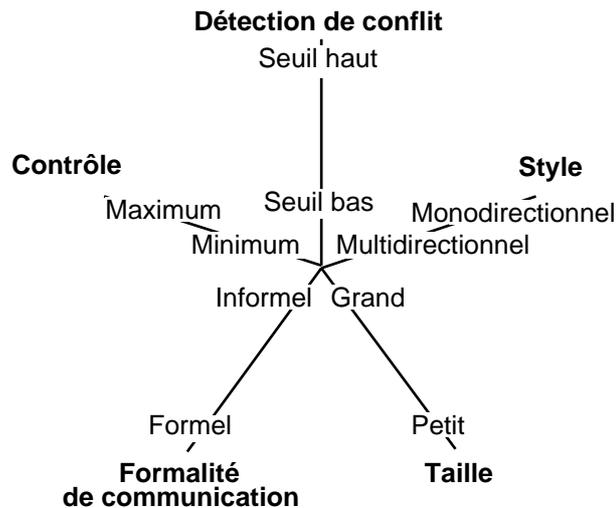


Figure 8
Protocole social
d'interaction

interaction synchrone (temps réel) ou asynchrone. Nous représentons ces cinq axes sur le diagramme de la Figure 7.

- **Protocole social d'interaction** (*Interactive social protocol*)
De tels modèles de protocoles sont analogues aux axes des situations d'interaction ci-dessus. Un protocole social se réfère aux séquences possibles d'échanges de signaux et d'information qui déterminent et identifient les discussions. Ceci inclut le concept d'hétérogénéité qui lui se réfère à la malléabilité des structures et des fonctions d'un groupe. Nous retrouvons ici cinq axes : la taille du groupe, les formalismes de communication, le contrôle des outils, la détection et la prise en compte des idées échangées (*contention detection* et *resolution*). La Figure 8 présente les axes qui caractérisent un protocole.

Toutefois, nous avons adapté ce modèle pour spécifier et concevoir le système CoVitesse [CoVitesse 2002] [Laurillau 2000] [Laurillau 1999b], système de navigation collaborative sur le WWW, dont une description de la mise en œuvre logicielle est présentée au Chapitre VII. C'est à l'aide de ce modèle, suite à une synthèse d'études ethnographiques, que nous avons mis en évidence et caractérisé quatre types de navigation collaborative qui sont offerts par le système CoVitesse. L'utilisation complète de ce modèle pour la conception du système CoVitesse est détaillée dans [Laurillau 1999a] et dans l'Annexe A.

GTA et analyse de la tâche

Jetant un pont entre les pratiques en IHM et l'ethnographie, la méthode GTA [Van Der Veer 1996] (*Groupware Task Analysis*) est un outil pour l'analyse de la tâche. Notons que l'analyse de la tâche est une étape incontournable dans la conception d'un système interactif : une structure hiérarchique des tâches est souvent obtenue [Coutaz 1990] [Shepherd 1989]. Plusieurs modèles de tâches sont construits au cours de la conception. GTA identifie trois modèles de tâches :

- *Modèle de tâches 1* : un système est motivé par la nécessité d'informatiser des pratiques de travail. Pour que le système soit conforme à ces méthodes de travail, il est nécessaire de capter et décrire les différentes tâches exécutées en situation réelle. Ceci fait l'objet de ce premier modèle de tâches qui capture les résultats de l'étude ethnographique. Ce premier modèle de tâches constitue un support à l'expression de besoins du système, définis au cours de la phase suivante.
- *Modèle de tâches 2* : cette seconde étape consiste à déterminer quels sont les besoins du système en se basant sur le premier modèle de tâches et aboutit à un second modèle de tâches. Il s'agit du point de vue système de la tâche, c'est-à-dire l'ensemble des tâches que l'utilisateur pourra effectuer avec le système. En passant à l'étape suivante, c'est-à-dire l'élaboration du modèle de la machine virtuelle, nous quittons l'étape d'analyse des besoins pour passer à l'étape de spécification du système. En ce sens, nous raisonnons à un niveau d'abstraction moins élevé.
- *Modèle de la machine virtuelle d'un utilisateur (user's virtual machine)* : cette dernière étape est l'élaboration d'un modèle de tâches "système" embarqué par le collecticiel selon un point de vue technologique. Il s'agit d'une description complète des tâches systèmes décomposées en actions physiques, c'est-à-dire la description des manipulations à réaliser avec l'interface pour exécuter une action. En ce sens, GTA couvre aussi en partie la phase de conception de l'espace IHM dans le cycle de vie en V. Ce modèle est un élément de spécification du système car ce modèle fait partie intégrante des spécifications externes et fonctionnelles. Ce point est abordé dans le paragraphe suivant.

Les formalismes pour décrire les arbres de tâches sont des formalismes orientés objet comme par exemple MAD [Scapin 1989] (Méthode Analytique de Description). De plus, pour élaborer ces modèles de tâches, GTA repose sur trois concepts :

- les participants : acteurs d'une tâche dans un rôle donné au sein d'une organisation,
- le travail : identifié par une tâche et structuré par un ensemble de sous-tâches et d'actions exécutées selon les règles sociales imposées,
- une stratégie : les tâches à réaliser en fonction d'un rôle donné,

- une situation de travail : identifiée par les objets manipulés et l'environnement de travail.

Ayant définis les besoins, nous quittons l'étape d'analyse des besoins pour passer à l'étape de conception et de spécification du système traitée dans le paragraphe suivant.

3.3. OUTILS POUR LES SPÉCIFICATIONS FONCTIONNELLES ET EXTERNES

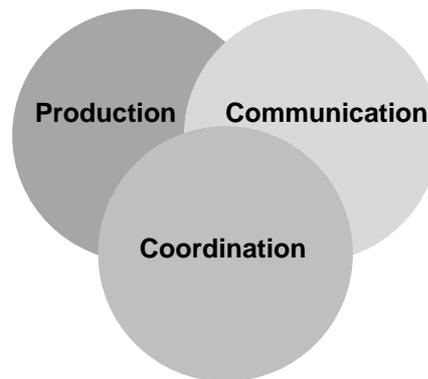


Figure 9

Modèle du trèfle [Salber 1995]

Ce modèle organise les fonctionnalités d'un collectif selon trois espaces : production, communication et coordination.

A partir de l'analyse de besoins, il convient alors de concevoir le modèle de l'interaction. Les spécifications externes d'un collectif s'expriment en termes de requis fonctionnels et requis utilisateurs.

Les requis fonctionnels cernent les fonctionnalités offertes par le collectif. Le modèle du trèfle, présenté dans la première section intitulée "Spécifications fonctionnelles et modèle du trèfle", propose un cadre pour organiser et définir les fonctionnalités d'un collectif.

Les requis utilisateurs concernent l'utilisabilité du collectif et donc l'interface du collectif. Communément, l'utilisabilité se décline en propriétés ergonomiques [Coutaz 2001]. Les propriétés ergonomiques propres aux collectifs sont présentées dans la seconde section intitulée "Spécifications externes et propriétés ergonomiques".

Spécifications fonctionnelles et modèle du trèfle

Le modèle du trèfle [Salber 1995], inspiré du modèle conceptuel d'un collectif proposé par C. Ellis [Ellis 1994], fournit un cadre conceptuel utile pour déterminer les requis fonctionnels et mener une analyse fonctionnelle. En effet, selon ce modèle présenté à la Figure 11, un collectif couvre trois espaces fonctionnels :

- **L'espace de production** concerne l'ensemble des fonctionnalités de production d'objets partagés tels que des documents communs et les fonctionnalités de réglementation des accès à ces données partagées.

Figure 10
InTouch [Brave 1998]

Le système InTouch est un jeu à retour d'effort collaboratif constitué de rouleaux fixés sur un socle. Dès qu'un rouleau subit une rotation, la force résultante est appliquée aux rouleaux correspondants des autres socles distants.



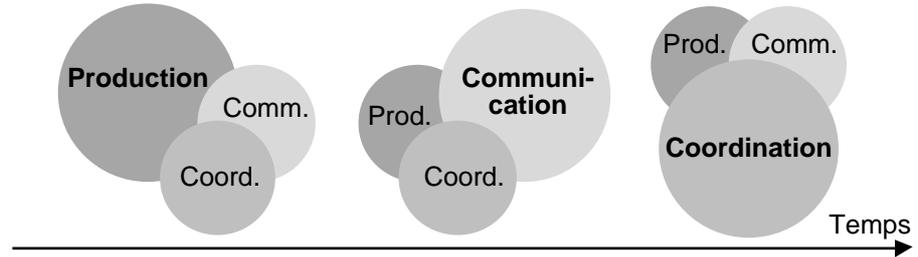
Par exemple, les éditeurs partagés, définis dans le paragraphe 2.1, sont dédiés à la production.

- **L'espace de communication** correspond aux fonctionnalités permettant l'échange d'information entre les acteurs du collectif. Cette échange est de la communication homme-homme médiatisée [Salber 1995] (CHHM). Il existe différents modes de communication comme, par exemple, l'audio (téléphone), la vidéo (*mediaspace*), le textuel (messageries), la gestuelle (langue des signes) ou l'haptique (communication à travers un système à retour d'effort comme, par exemple, le système InTouch [Brave 1998] présenté à la Figure 10). Les systèmes de vidéoconférence et les *mediaspace*, définis dans le paragraphe 2.1, sont dédiés à la communication homme-homme médiatisée.
- **L'espace de coordination** correspond aux fonctionnalités dédiées à l'assignation de tâches et de rôles aux différents acteurs d'une activité collaborative. Ces fonctionnalités ont pour but de coordonner les acteurs afin de réaliser une œuvre commune. Cette coordination peut s'exprimer en terme de planification de tâches. Par exemple, les systèmes *workflow*, définis dans le paragraphe 2.1, sont des systèmes dédiés à la planification de transferts de documents entre différents intervenants au cours d'un processus industriel.

Les trois espaces fonctionnels d'un collectif étant définis, il convient néanmoins de noter que certaines fonctionnalités peuvent être à l'intersection de plusieurs espaces comme le souligne la Figure 9 (intersection entre les trois espaces). Ainsi, la distinction selon les trois espaces n'est pas stricte car il est tout à fait possible qu'une activité de coordination ait lieu suite à une activité de communication. Par exemple, la prise de rendez-vous par téléphone est une activité de coordination

basée sur une activité de communication. Ce point est étudié plus en détail dans les chapitres suivants.

Figure 11
Evolution dans le temps du rôle fonctionnel d'un collectif



De plus, le rôle fonctionnel global d'un collectif, comme le montre la Figure 11, peut évoluer au cours du temps : cette évolution peut être modélisée grâce au modèle du trèfle. Par exemple, lors de l'utilisation d'un système dédié à la production, il est possible, à un moment donné, que l'activité de groupe soit centrée sur la communication en vue de se coordonner afin de redéfinir l'activité de production. Cette approche permet de prendre en compte la variabilité dans les activités de groupe au cours du temps.

Comme nous l'avons évoqué, le modèle du trèfle est un modèle de conception utile pour déterminer et organiser les fonctionnalités d'un collectif selon la production, la communication et la coordination. Par exemple, F. Tarpin propose une méthode pour spécifier les fonctionnalités d'un collectif en fonction d'un ensemble de questions thématiques, en s'appuyant sur les trois espaces fonctionnels du modèle du trèfle [Tarpin 1997] :

- **Production** : structuration et gestion des données, par identification des données partagées et de leurs structures, par modélisation du support du travail et des opérations de production et par identification des caractéristiques de partage.
- **Coordination** : organisation du processus de travail par identification des tâches et modélisation du processus de travail et par identification des rôles et des modes de coopération (couplage).
- **Communication** : choix des modes de communication et des protocoles de conversation (rôles et attribution de responsabilités).

Spécifications externes et propriétés ergonomiques

Le document de spécifications externes décrit le système tel qu'il sera perçu et utilisé par l'utilisateur. Ces spécifications détaillent le modèle d'interaction proposé à l'utilisateur pour qu'il puisse interagir avec le système.

Comme toute application logicielle, la production d'un collectif doit répondre à des exigences et à un niveau de qualité requis. En génie logiciel, il existe de nombreux facteurs [McCall 1977] pour quantifier la

qualité du logiciel comme par exemple la modifiabilité du logiciel. Dans le domaine de l'IHM, les efforts se concentrent plus particulièrement sur l'utilisabilité d'un système interactif qui est un facteur de qualité [McCall 1977]. Ce facteur définit la qualité d'utilisation d'un système interactif et sert de guide pour la conception et l'évaluation. Nous détaillons ce dernier point dans le paragraphe consacré à l'évaluation ergonomique. Une approche en IHM [IFIP 1996] est de décliner ce facteur selon trois facteurs qualité : la facilité d'apprentissage (facilité d'apprentissage pour un utilisateur novice à manipuler le système), la souplesse (capacité du système à proposer un éventail de choix à l'utilisateur pour exécuter une action) et la robustesse de l'interaction (capacité du système à faciliter l'accomplissement d'une action). McCall affine les facteurs en critères ; ici les trois facteurs sont affinés en un ensemble de propriétés ergonomiques. Par exemple, l'adaptativité est une propriété ergonomique qui est liée au facteur de souplesse et traduit la capacité du système à s'adapter à l'utilisateur sans une intervention explicite de sa part [IFIP 1996]. L'ensemble de ces propriétés guide le concepteur dans son activité de conception et notamment pour les spécifications externes du système.

Dans le cas des collecticiels, de nouvelles propriétés ergonomiques sont spécifiques à l'activité de groupe. Nous présentons les propriétés ergonomiques les plus étudiées issues de [Salber 1995] [UsabilityFirst 2002] :

- **Conscience de groupe et rétroaction de groupe** (*group awareness* et *feedthrough*) : lors d'une activité en groupe, par un exemple un sport tel que le basket, les joueurs ont toujours un oeil sur ce que font les autres membres du groupe pour agir et réagir en fonction des tâches réalisées (et de leurs résultats). Le groupe est efficace, en se basant encore sur la métaphore sportive, que s'il existe un esprit d'équipe qui s'appuie sur la conscience de groupe. Il s'agit d'informer un utilisateur de l'activité en cours des autres utilisateurs. Le terme employé dans la littérature est la **conscience de groupe**. Cette source d'information n'est pas essentielle à la réalisation de la tâche en cours mais elle y contribue dans le sens où il s'agit d'informations relatives à l'état d'avancement de l'activité des autres utilisateurs. Par exemple, les *mediaspace*, définis dans le paragraphe 2.1, sont une classe de collecticiels où la notion de conscience de groupe est primordiale. En effet, le but principal est de maintenir une conscience de groupe forte entre des individus dispersés géographiquement. Néanmoins, les *mediaspace* soulèvent de nombreux problèmes liés à la protection de l'espace privé (propriété abordée au point suivant) de par la présence d'une caméra dans les bureaux et espaces communs.

La propriété de conscience de groupe traduit donc la capacité du

système à rendre observable des informations sur l'activité de groupe indépendamment de la tâche en cours.

Une propriété liée à la conscience de groupe est celle de **rétroaction de groupe** qui traduit la capacité du système à rendre observable des informations sur l'activité de groupe pertinente pour la réalisation de la tâche en cours. Cette propriété est liée au WYSIWYS abordé dans la suite.

- **Protection de la vie privée (*privacy*)** : la propriété de protection de la vie privée est relative à la protection des informations privées [UsabilityFirst 2002]. De nombreuses propriétés sont en relation avec celle-ci, à savoir la propriété d'observabilité et de publication, et la propriété de réciprocité et d'identification. La conception d'un collecticiel impose un bon dosage entre la propriété de protection de la vie privée et la propriété de conscience de groupe.
- **Observabilité et publication** : la propriété d'observabilité n'est pas nouvelle puisqu'elle est considérée pour les systèmes interactifs mono-utilisateurs [IFIP 1996] (page 84). Pour les systèmes interactifs, l'observabilité traduit la capacité du système à rendre observable son état interne et à offrir des moyens à l'utilisateur pour qu'il puisse le consulter. Dans le cas des collecticiels, cette notion d'observabilité s'applique aussi aux informations concernant les autres utilisateurs et leurs activités. D. Salber [Salber 1995] désigne cette propriété par la notion d'observabilité publiée. Des informations observables de l'utilisateur, nous considérons trois formes d'observabilité [Laurillau 1999a] :
 - * **Observable** : ce qui doit être perçu par tout utilisateur,
 - * **Publiable** : ce qu'un utilisateur décide de rendre observable de lui-même,
 - * **Filtrable** : la forme de ce qui est observable. Les informations ainsi filtrées donnent de l'utilisateur l'apparence qu'il a choisie.

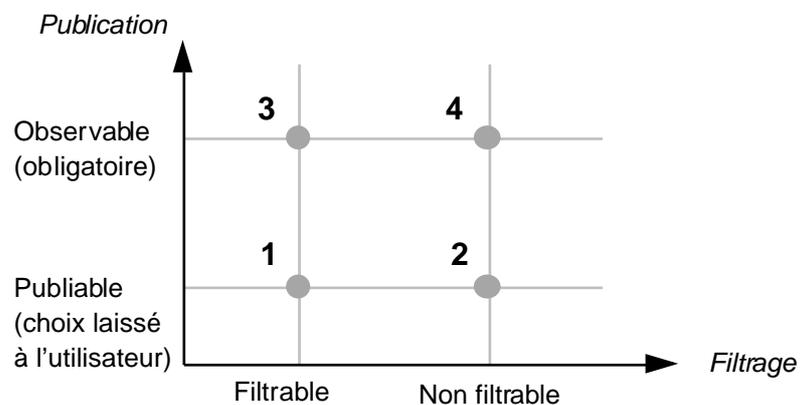


Figure 12
Quatre niveaux d'observabilité
[Laurillau 1999a]

La différence entre un filtrage et une information publiée se situe dans la véracité des informations : quand une information est publiée, celle-ci est a priori exacte, quand celle-ci est filtrée, elle est modifiée, elle est modelée à sa propre façon.

Nous définissons [Laurillau 1999a] quatre niveaux d'observabilité, présentés à la Figure 12 : les quatre niveaux sont définis selon deux axes orthogonaux, *Filtrage* et *Publication*. Pour chaque entité, le concepteur doit décider quel niveau d'observabilité est nécessaire pour que l'utilisateur puisse réaliser sa tâche tout en préservant son espace privé et celui des autres. Ainsi, un moyen possible de protéger son espace privé est de doter le système de filtres des informations paramétrables par l'utilisateur. Par exemple, dans le *mediaspace Comedi* [Coutaz 1997], un ensemble de filtres de l'image vidéo est proposé pour rendre compte de l'activité dans un bureau tout en préservant l'intimité des utilisateurs. Parmi les filtres proposés, l'utilisateur peut choisir de masquer son image à l'aide d'un "store vénitien". Ce filtre tout en masquant les personnes permet néanmoins de deviner l'activité dans le bureau, comme le nombre de personnes. Dans le système CoVitesse, présenté dans le Chapitre VII, le système propose ensemble de filtres pour ne rendre observable qu'une partie des pages web visitées. Un de ces filtres permet de masquer toutes les pages visitées contenant un mot particulier, ce dernier définissant le filtre.

- **Réciprocité (*reciprocity*) et identification** : les propriétés de réciprocité et d'identification sont des cas particuliers de l'observabilité. **La propriété de réciprocité** exprime la capacité du système à fournir des moyens pour que des utilisateurs puissent s'observer mutuellement. Ainsi, si un utilisateur obtient des informations sur un autre utilisateur, ce dernier doit pouvoir obtenir le même type d'information sur le premier. Par exemple, si un utilisateur peut observer un autre utilisateur à travers une vidéo, ce dernier doit aussi pouvoir l'observer. **La propriété d'identification** traduit la capacité du système à offrir des moyens pour qu'un utilisateur puisse identifier les auteurs d'une action ou d'une décision.
- **WYSIWIS (*What You See Is What I See*) et couplage de l'interaction (*coupling*)** : WYSIWIS signifie en français "ce que tu vois est ce que je vois". Dans la suite du manuscrit, nous utilisons systématiquement l'abréviation WYSIWIS. La **propriété de WYWISIS** se traduit en pratique par une vue identique entre plusieurs utilisateurs ; dès qu'un utilisateur apporte une modification à la vue courante (par exemple, le déplacement d'une barre de défilement). Cette notion ne se limite pas à l'interface car cette

propriété impose que toutes modifications, que ce soit au niveau de l'interface ou au niveau fonctionnel, soient diffusées à tous les autres utilisateurs.

La modification apportée à la vue courante est répercutée dans les autres vues avec un délai plus ou moins important suivant le **degré de couplage** (fortement couplé ou faiblement couplé). Il existe deux modes *WYSIWIS* qui ont un impact direct sur la nature du couplage de l'interaction : le mode *WYSIWIS* strict et le mode relâché. Dans le premier cas, cela signifie que l'interaction est fortement couplée et que tous les utilisateurs disposent nécessairement d'une vue unique. En l'occurrence, l'espace privé est très limité. Par exemple, certains collecticiels ne disposent que d'un seul pointeur de souris soumis à une politique de partage. Dans le second mode, l'interaction est plus souple et les utilisateurs disposent de leur propre vue et de leur propre espace privé.

Cette propriété exprime une tension entre la propriété de protection de la vie privée et la conscience de groupe. En effet, un mode *WYSIWIS* strict aura tendance à favoriser la conscience de groupe (par exemple il n'y a qu'un seul pointeur et tous les utilisateurs sont informés de la manipulation du pointeur) tandis que le mode *WYSIWIS* relâché aura tendance à favoriser la protection de l'espace privé (par exemple l'utilisateur dispose de son propre pointeur et travaille dans son espace privé mais n'est pas nécessairement informé de l'activité des autres).

- **Viscosité** : la viscosité est un phénomène social qui exprime l'incidence d'une action, c'est-à-dire les effets secondaires d'une action, sur l'activité des autres utilisateurs. Ces effets secondaires peuvent se traduire par une surcharge de travail pour les autres utilisateurs par l'apparition de tâches supplémentaires. Il s'agit d'une forme particulière de la propriété de couplage de l'interaction. Par exemple, dans un éditeur de dessin partagé, si un utilisateur décide de déplacer la zone de dessin commune avec une barre défilement collaborative, les autres utilisateurs sont obligés de s'interrompre et de recentrer leur outil de dessin pour se replacer au bon endroit et reprendre leurs activités.

3.4. OUTILS POUR LA CONCEPTION LOGICIELLE

Tandis que dans le paragraphe précédent nous avons proposé un ensemble d'outils pour la conception ergonomique du collecticiel, nous nous intéressons à la conception logicielle en faisant l'hypothèse que le collecticiel est conçu et spécifié et qu'il convient maintenant de le développer.

Le développement du logiciel se déroule en deux grandes étapes, comme indiqué à la Figure 7 : la phase de conception logicielle, qui se décompose elle-même en deux sous étapes de conception globale et détaillée, et la

phase de réalisation logicielle (codage et tests unitaires). Cette approche est commune au développement de tout système interactif mais est encore plus complexe dans le cas des collecticiels. L'étape de conception globale est donc une étape charnière puisqu'elle a la charge de faire le lien entre l'espace IHM et l'espace logicielle. C'est au cours de cette étape qu'il convient de construire une architecture logicielle. Il existe deux types d'architecture logicielle : l'architecture conceptuelle et l'architecture implémentationnelle. La première décrit une organisation des fonctionnalités en modules et décrit les protocoles de communication entre ces modules. La seconde est l'implémentation (i.e. la version programmée) de l'architecture conceptuelle qui est dépendante des outils de codage. Ce dernier point est abordé dans le paragraphe suivant.

Pour concevoir l'architecture conceptuelle du système, il convient d'utiliser un modèle d'architecture : l'usage commun assimile une architecture conceptuelle à un ensemble organisé de composants dont les interactions sont médiatisées par des entités spécialisées ou connecteurs. Le processus de conception d'une architecture conceptuelle recouvre les activités suivantes : définition de la décomposition fonctionnelle du système, identification de son organisation structurelle, allocation des fonctions à la structure et définition de la coordination entre les entités de la structure. Le résultat de ces quatre activités constitue la structure modulaire ou architecture conceptuelle.

Les modèles d'architecture pour collecticiels sont principalement conceptuels comme Clock [Graham 1996], PAC* [Calvary 1997] ou le modèle de Dewan [Dewan 1999]. Ces modèles définissent des structures modulaires normalisées. Par exemple, le modèle de Dewan préconise un empilement de couches partagées et répliquées communiquant entre elles par échanges d'événements.

Comme nous l'avons évoqué, avec l'architecture logicielle conceptuelle nous quittons l'espace de conception proprement dit du système interactif pour pénétrer dans l'espace logiciel réservé aux informaticiens. La transition de l'espace IHM à l'espace logiciel est toujours une étape délicate puisqu'il s'agit de transcrire les concepts identifiés au cours de l'analyse des besoins et de la conception de l'interface sous forme d'une organisation de fichiers, d'algorithmes et de structures de données. Cette situation charnière [Nigay 1997] se traduit par une tension permanente entre d'une part les contraintes techniques imposées par les outils de mise en œuvre et d'autre part la satisfaction des requis de l'utilisateur, des spécifications externes de l'IHM et des spécifications fonctionnelles du système. Aussi, il est important que les modèles d'architecture conceptuelle véhiculent, outre des propriétés logicielles, des propriétés ergonomiques et intègrent des éléments de conception du système interactif.

La notion d'architecture logicielle, conceptuelle et implémentationnelle, et de modèle d'architecture pour les collecticiels fait l'objet du chapitre suivant.

Après avoir conçu l'architecture logicielle conceptuelle dans le cycle de vie, il convient ensuite de concevoir l'architecture implémentationnelle puis de développer le système. Nous étudions au paragraphe suivant les outils disponibles pour cette phase de codage.

3.5. OUTILS POUR LA RÉALISATION LOGICIELLE

L'étape de la réalisation logicielle du collecticiel consiste à traduire l'architecture conceptuelle, issue de l'application d'un modèle, en une architecture implémentationnelle puis de développer le système.

Actuellement, le codage des applications logicielles ne peut se faire sans l'utilisation d'outils de développement dans le but de simplifier la tâche du développeur et de lui éviter de tout réinventer. Cependant, les outils ne font pas tout car ils offrent essentiellement des services élémentaires et la majeure partie du travail est à la charge du développeur. Ces outils sont constitués d'un capital de fonctionnalités que le développeur peut utiliser et assembler pour produire le système. Dans le cas des collecticiels, les outils de développement deviennent indispensables étant donné la complexité de mise en œuvre de ces systèmes (accès concurrents, perte d'information sur le réseau, etc). Dans ce paragraphe, nous présentons deux types d'outils : les services logiciels, et les boîtes à outils et infrastructures logicielles. Les services logiciels sont des regroupements de fonctionnalités capables de choisir l'algorithme le plus adapté à une situation donnée, réduisant ainsi la charge de développement. Par exemple, dans le cas des collecticiels, un service de verrouillage des données doit être capable de choisir entre un algorithme de verrouillage optimiste ou pessimiste en fonction de la situation. Les boîtes à outils et infrastructures logicielles reposent sur un ensemble de services permettant ainsi le développement d'applications.

La première section présente un ensemble de services logiciels spécifiques aux collecticiels. La seconde section traite ensuite des boîtes à outils et des infrastructures logicielles qui reposent sur un ensemble de services.

Services logiciels pour les collecticiels

Le développement logiciel d'un collecticiel repose sur des services logiciels. De nombreux services généraux ont été identifiés pour mettre en œuvre la collaboration. Bien sûr, un collecticiel n'exploite pas nécessairement tous les services. Nous listons ci-dessous les services logiciels les plus fréquemment traités dans la littérature du domaine [Dewan 2001] [UsabilityFirst 2002] :

- **Gestion de sessions** (*session management*): le gestionnaire de sessions est un service qui maintient à jour la liste des sessions et des différents participants. Une session est une période pendant laquelle les utilisateurs participent à l'activité de groupe. L'accès à une session en cours est régi par des droits d'accès. C'est à ce niveau que le gestionnaire met en place une politique de gestion des participants rejoignant une session en cours (*latecomers*) en donnant la possibilité de rejouer les actions réalisées dans le passé.
- **Contrôle de la concurrence** (*concurrency control*): ce service a pour rôle de veiller à ce que des actions concurrentes puissent accéder aux objets partagés et que l'état du système reste cohérent. Une technique courante consiste à verrouiller (*locking*) les objets partagés. Notamment, il existe de nombreux mécanismes tels que le verrouillage optimiste ou pessimiste [Prakash 1999].
- **Contrôle d'accès** (*access control*): ce service gère les droits d'utilisation des outils et le droit d'exécution de certaines fonctionnalités. Les droits d'accès sont définis en fonction des rôles assignés aux différents participants. Le mécanisme de "contrôle d'utilisation" (*floor control*) a la charge de répartir les plages d'utilisation d'un outil. Par exemple, il peut s'agir du droit d'utilisation d'un tableau blanc partagé par une seule personne pendant un certain laps de temps. Ce mécanisme de "contrôle d'utilisation" existe sous plusieurs formes comme, par exemple, le mécanisme de prise de parole (*turn-taking*).
- **Comparaison et fusion de données** (*diffing et merging*): ces mécanismes, souvent associés aux éditeurs partagés asynchrones, permettent de comparer et de fusionner différentes versions d'un même document. Il existe de nombreux algorithmes de fusion dont l'algorithme de fusion divergente avec garantie de consistance de P. Dourish [Dourish 1996a]: cet algorithme autorise plusieurs versions d'un document à un instant donné, tout en garantissant que la fusion ultérieure des différentes versions sera cohérente.
- **Défaire-refaire** (*undo-redo*): ce service, classique dans les systèmes interactifs mono-utilisateurs, est complexe à mettre en œuvre dans le cas des collecticiels. En effet, ce service doit veiller à ce que les actions concurrentes de défaire-refaire n'entraînent pas un état incohérent ou non désiré du système. Par exemple, dans le cas d'un éditeur de dessin partagé, un utilisateur décide d'annuler la dernière action, c'est-à-dire le dessin d'un cercle. Au même moment, un autre utilisateur décide de dessiner un rectangle. Ce service doit alors veiller à ce que le cercle

soit supprimé au lieu du rectangle, sinon l'état obtenu est incompatible avec l'état désiré. Ce service doit respecter exactement dans quel ordre les actions ont été réalisées. Par exemple, différents modèles et algorithmes pour mettre en œuvre ce service sont décrits dans [Choudary 1995].

- **Synchronisation des présentations** (*coupling*) : ce service a la charge de synchroniser toutes les instances de la présentation du collectif en propageant les événements graphiques. Cette dépendance entre les différentes vues est plus ou moins forte selon le degré de couplage (*WYSIWIS* strict ou relâché). Par exemple, si un utilisateur, dans le cas d'un collectif disposant d'un unique pointeur de souris partagé (*WYSIWIS* strict), déplace sa souris, alors le pointeur doit se déplacer automatiquement au niveau de toutes les autres vues. De nombreux travaux portent sur l'étude du couplage des interfaces [Dewan 1995].
- **Notification** : le service de notification a la charge de propager les changements d'état du système et d'informer les différents clients de ces changements d'état. Ce service est similaire au précédent mais celui-ci relève du Noyau Fonctionnel. Ce service participe au maintien de la conscience de groupe. D. Ramduny et A. Dix [Ramduny 1998] ont notamment étudié les apports de ce service et les différentes configurations pour le mettre en œuvre.

Boîtes à outils et infrastructures logicielles

Les outils de développement sont les boîtes à outils pour les collecticiels (*groupware toolkit*) et les infrastructures, deux types d'outils qui s'appuient sur des services comme ceux exposés dans la section précédente. Nous définissons une infrastructure comme une boîte à outils pour les collecticiels imposant un schéma d'exécution statique ou dynamique. Les outils de développement de collecticiels sont présentés et analysés dans le Chapitre V. Citons ici, à titre d'exemple, la boîte à outils GroupKit [Roseman 1992] qui est certainement l'outil de développement de collecticiels le plus connu dans le monde du TCAO. Cette boîte à outils met à disposition du développeur de nombreuses fonctionnalités comme, par exemple, le pointeur de souris collaboratif. Ce pointeur est un objet graphique qui répond au mouvement de la souris d'un utilisateur. Ainsi dès que l'utilisateur bouge sa souris, ce pointeur est déplacé dans toutes les vues. Il s'agit ici d'un exemple de service, en l'occurrence un service de couplage de l'interaction, que la boîte à outils intègre : ce service évite donc au développeur de reconcevoir cette fonctionnalité qui aurait nécessité de programmer la gestion des différentes vues du pointeur et de programmer l'association entre le déplacement de la souris et le déplacement du pointeur dans toutes les vues.

Pour conclure sur les outils de réalisation logicielle de collecticiels, il est important de souligner le fait que ces outils, comme GroupKit [Roseman 1992] ou COCA [Li 1999], n'éliminent pas la nécessité d'une architecture conceptuelle élaborée lors de l'étape précédente de conception logicielle. L'architecture conceptuelle est un artefact incontournable tant que les outils impliquent une phase de programmation. Sans une architecture conceptuelle, le collecticiel est plus difficile à développer, à modifier, à étendre et à maintenir. Une grande difficulté est de disposer d'outils qui offrent un haut niveau d'abstraction adapté pour réduire à la fois le coût de développement et pour faciliter le passage de l'architecture conceptuelle à la réalisation logicielle. L'architecture conceptuelle intégrant des concepts issus de l'espace IHM (Figure 7) en termes de composants logiciels, les outils doivent donc permettre d'exprimer à un coût réduit ces concepts liés à l'interaction.

3.6. OUTILS POUR L'ÉVALUATION ERGONOMIQUE

L'évaluation ergonomique intervient à toutes les étapes du cycle de vie logiciel, quel que soit le type de système interactif (mono-utilisateur ou collecticiel). L'évaluation ergonomique permet de détecter les problèmes d'utilisabilité du collecticiel a priori lors des phases de conception ou a posteriori, lorsque le système interactif a été développé. Dans ce dernier cas, les résultats de cette évaluation expérimentale peuvent donner lieu à une nouvelle itération du cycle de conception.

Il existe donc deux types de techniques pour l'évaluation, les techniques prédictives (a priori) et les techniques expérimentales (a posteriori). Le premier type de techniques repose sur des modèles théoriques tels que le modèle *ICS* [Barnard 1987] (*Interactif Cognitive Subsystems*, modèle cognitif de l'utilisateur) ou des modèles de performance comme, par exemple, le modèle *GOMS* [Card 1983] (*Goals, Operators, Methods and Selection Rules*), ainsi que l'utilisation d'heuristiques comme celles proposées par J. Nielsen [Nielsen 1990]. Ces heuristiques peuvent être liées à des propriétés ergonomiques, comme celles que nous avons exposées au paragraphe 3.3. Le second type de techniques repose sur une évaluation expérimentale du système à partir de prototypes, de maquettes ou du produit final, par observation des utilisateurs en situation d'utilisation. Notons enfin la technique du magicien d'oz [Dahlbäck 1993] qui permet d'obtenir des résultats expérimentaux alors que le système n'est pas développé : il s'agit donc d'une technique prédictive qui consiste à simuler le comportement du système, à l'insu de l'utilisateur observé, par un compère humain. A l'issue d'une séance, les observations sont souvent complétées par un questionnaire.

L'évaluation d'un collecticiel est nettement plus complexe que l'évaluation d'un système interactif mono-utilisateur. En effet il est difficile et coûteux de mener une évaluation car cela nécessite de disposer

d'un grand nombre d'utilisateurs et de former des groupes homogènes. Or de nombreux critères doivent être pris en compte pour caractériser un groupe : la taille, la représentativité sociale, le niveau de compétence dans le domaine de la tâche, le niveau de compétence d'utilisation des outils informatiques, etc. Les techniques d'évaluation des systèmes mono-utilisateur doivent donc être adaptées afin de traiter ces nouveaux paramètres.

Or les travaux dans ce domaine sont pauvres. Dans la littérature, nous avons relevé une technique basée sur des heuristiques [Backer 2001]. Visant à adapter aux collecticiels la technique d'évaluation reposant sur des heuristiques de J. Nielsen [Nielsen 1990], K. Backer propose un ensemble de huit heuristiques issues de la théorie de la mécanique de la collaboration [Gutwin 2000]. Cette dernière identifie six actions de base qui régissent la mécanique de collaboration dans une situation d'espace partagé : communication, coordination, planification, consultation (*monitoring*), assistance et protection. Les huit heuristiques identifiées à partir de ces actions de base sont alors les suivantes

- 1 Un collecticiel doit fournir des moyens pour rendre observable l'échange verbal d'informations.
- 2 Un collecticiel doit fournir des moyens pour rendre observable l'échange par la gestuelle d'informations.
- 3 Un collecticiel doit fournir des moyens pour rendre observable l'échange d'informations induit par le comportement (position du corps, des mains, des yeux, expression du visage, etc.).
- 4 Un collecticiel doit fournir des moyens pour rendre observable l'échange d'informations induit par la manipulation d'artefacts partagés (manipulation d'objets partagés et retour d'information sur les changements d'états de ces objets).
- 5 Un collecticiel doit fournir des moyens pour protéger l'espace de travail, l'espace privé, les objets partagés, etc.
- 6 Un collecticiel doit permettre plusieurs modes d'interaction (fortement couplé ou faiblement couplé) et la possibilité de modifier dynamiquement le niveau de couplage.
- 7 Un collecticiel doit mettre en œuvre des moyens pour autoriser les participants à coordonner leurs actions.
- 8 Un collecticiel doit permettre aux utilisateurs de découvrir qui sont tous les participants et doit faciliter le contact entre ces participants.

Les heuristiques sont employées pour diagnostiquer des problèmes potentiels d'utilisabilité liés à une interface. La méthode est très simple et consiste à faire inspecter l'interface par des évaluateurs selon les

heuristiques. L'évaluateur commente l'interface pour chaque heuristique. Couramment, cette inspection nécessite entre trois et cinq évaluateurs maximum qui ont la capacité de détecter entre 75 et 80% des problèmes d'utilisabilité [Gutwin 2000]. Cette approche a l'avantage d'offrir une méthode d'évaluation à faible coût qui très populaire dans les milieux industriels. Celle-ci peut être appliquée par des évaluateurs qui ne sont pas experts car les heuristiques sont souvent bien documentées.

3.7. CONCLUSION

Dans cette partie du chapitre, nous avons présenté un ensemble de méthodes, modèles et outils pour la conception et la réalisation d'un collecticiel. Pour les présenter et bien cerner leurs portées, notre contribution a été de les situer par rapport aux étapes du cycle de vie du logiciel en V. Néanmoins cette partie ne constitue pas un état de l'art des outils existants. Notre objectif était ici de présenter un panorama des outils, organisé dans un canevas fédérateur, afin de mieux situer notre étude dont nous rappelons les objectifs dans la partie suivante.

4. Conclusion

**4.1. MISE EN ŒUVRE
DES COLLECTICIELS :
OBJECTIFS DE
L'ÉTUDE**

Dans ce chapitre, nous avons d'abord souligné l'ampleur de l'espace des possibilités pour les collecticiels. Nous avons ensuite rappelé, par la présentation d'outils et de méthodes de mise en œuvre des collecticiels que le processus est organisé selon deux principales classes d'étapes de conception : les étapes de conception IHM (espace IHM) et les étapes de conception logicielle (espace Logiciel).

Les sciences humaines nourrissent essentiellement l'espace IHM grâce à l'apport d'outils et de méthodes issus de l'ethnologie ou de la sociologie (GTA), de modèles de l'utilisateur de psychologie cognitive et de modèles pour l'analyse fonctionnelle (modèle du trèfle). L'objectif affiché de ces outils, méthodes et modèles est d'aider à obtenir une spécification du collecticiel répondant aux besoins, identifiés au cours de l'analyse des besoins, en termes de spécifications fonctionnelles et de spécifications externes vérifiant des propriétés ergonomiques.

Les étapes suivantes consistent à exprimer en termes logiciels les spécifications : ce sont les étapes de l'espace Logiciel. C'est à l'aide des outils de conception logicielle, tels que les modèles d'architecture, et des outils de développement, tels que les boîtes à outils, que le développeur conçoit et code le logiciel du collecticiel.

Néanmoins, la transition entre l'espace IHM et l'espace Logiciel est complexe et délicate car il s'agit de traduire des spécifications sous forme de structures de données et d'algorithmes exécutés par des processus. Avec les étapes d'analyse des besoins et de conception, nous quittons l'espace de conception proprement dite de l'IHM pour pénétrer dans l'espace Logiciel réservé aux informaticiens. Le passage d'un espace à l'autre n'est pas sans difficulté [Nigay 1997] : le concepteur et le développeur ont l'obligation de satisfaire les spécifications fonctionnelles et externes de l'IHM tout en tenant compte des contraintes techniques imposées par les outils de mise en œuvre. Existe-t-il ainsi des outils logiciels pour la conception et le développement logiciels de collecticiels adaptés pour accompagner cette transition et permettre, entre autre, de satisfaire les requis ergonomiques tout en facilitant la réalisation logicielle ?

Dans ce contexte, l'objectif de cette étude est :

- d'analyser les outils logiciels existants au regard de cette transition de l'espace IHM vers l'espace Logiciel,
- de définir des outils de conception et de réalisation logicielles qui soient complémentaires à ceux existants et adaptés pour accompagner cette transition.

Nous nous sommes donc intéressés aux modèles d'architecture logicielle ainsi qu'aux boîtes à outils et infrastructures pour le développement des collecticiels. Ainsi le Chapitre III dresse un état de l'art des modèles d'architecture logicielle pour collecticiels tandis que le Chapitre IV présente notre contribution, le modèle d'architecture Clover pour faciliter la transition entre l'espace IHM et l'espace Logiciel. Symétriquement, le Chapitre V propose un état de l'art des boîtes à outils et infrastructures et le Chapitre VI notre contribution, la plate-forme Clover. Le Chapitre VII illustre l'utilisation de la plate-forme Clover en détaillant la mise en œuvre du système CoVitesse, un logiciel de navigation collaborative sur le WWW.

4.2. DÉMARCHE DE TRAVAIL

La démarche de travail adoptée dans cette étude se place du point de vue de l'activité de groupe. La position ainsi retenue apporte une dimension novatrice et complémentaire par rapport aux travaux existants. Comme nous l'avons évoqué dans le paragraphe précédent, l'objectif de cette étude porte sur la transition entre l'espace IHM et l'espace Logiciel. Par conséquent, il convient de se positionner par rapport à la phase de spécification, c'est-à-dire par rapport aux paramètres des spécifications fonctionnelles et externes, en se basant sur le modèle du trèfle et les propriétés ergonomiques relatives aux collecticiels (présentés au

paragraphe 3.3). Aussi, nous proposons une grille d'analyse qui organise ces paramètres au sein d'un canevas intégrateur présenté à la Figure 13.

Cette grille est ensuite utilisée tout au long de ce manuscrit pour étudier les architectures conceptuelles et les outils de développement pour les collecticiels mais aussi pour situer nos contributions. Nous verrons qu'il y a plusieurs niveaux de compatibilité entre les éléments de la grille et les modèles d'architecture et outils de développement. Nous reprenons ici les trois niveaux de compatibilité identifiés dans [Nigay 2001]. Selon les modèles et outils, les éléments de la grille seront explicites par localisation dans un composant logiciel (faible compatibilité), seront liés à des motifs architecturaux (compatibilité moyenne) ou encore seront associés à des services généraux ou mécanismes architecturaux (forte compatibilité). Plusieurs travaux en IHM pour des systèmes mono-utilisateurs étudient cette compatibilité entre les concepts de l'IHM et la réalisation logicielle : outre l'approche de notre équipe IIHM [Nigay 2001] qui est dépendante du modèle d'architecture logicielle PAC-Amodeus [Nigay 2001], nous relevons les travaux récents de L. Bass et B. John [Bass 2001] qui visent à établir des couples, scénarios d'utilisabilité liés à des propriétés ergonomiques et mécanismes architecturaux, ces travaux étant indépendants d'un modèle d'architecture logicielle particulier.

La grille d'analyse, centrée sur l'activité de groupe, repose sur le modèle du trèfle et les propriétés ergonomiques, deux outils présentés au paragraphe 3.3 pour la conception ergonomique de collecticiels. Pour définir la structure de la grille, nous exploitons la notion de contexte supposé partagé des actions, sachant qu'une action peut être individuelle ou collective [Hoogstoel 1995].

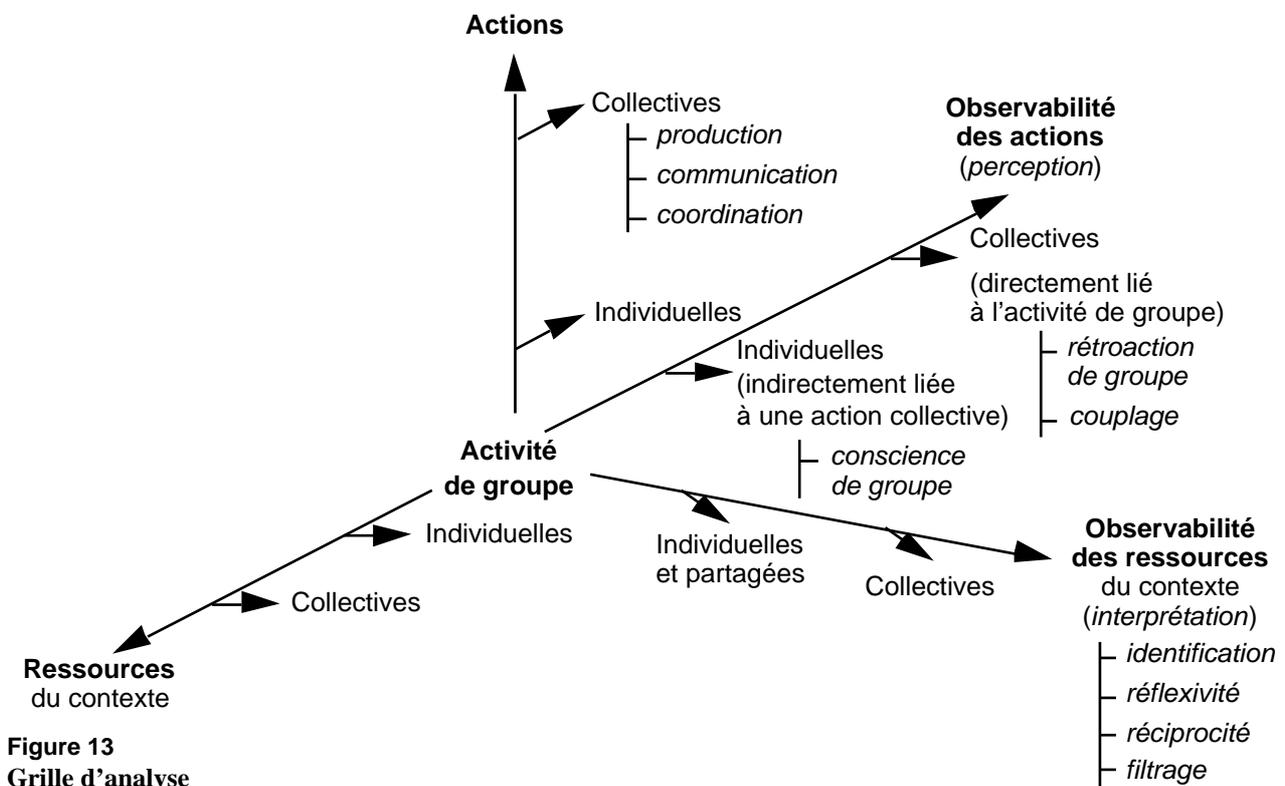
Le contexte, selon L. Karsenty [Karsenty 1997], est l'ensemble des ressources disponibles et exploitées par un utilisateur pour interpréter et exécuter une action. Il existe deux types de contexte, le contexte individuel et le contexte partagé. Précisément, le contexte est **supposé** partagé puisqu'il représente une "*compréhension mutuelle des actions de chacun*" et l'articulation des actions "*suppose un certain degré de contexte partagé (l'intersection entre les contextes individuels)*". Ce contexte est nécessairement "supposé" partagé puisque chacun dispose de sa propre interprétation du contexte partagé ce qui peut induire dans certaines situations des incompréhensions.

Selon L. Karsenty [Karsenty 1997], le contexte joue un rôle important dans la mise en œuvre de l'activité de groupe. En effet, l'activité de groupe résulte d'une articulation d'actions dans un contexte donné et il apparaît que cette articulation repose essentiellement sur la communication et l'exploitation de données contextuelles.

Le contexte est constitué de six niveaux de ressources organisés en deux grandes familles :

- La première est liée à la connaissance et à la cognition : cette forme de contexte est relative à la culture, à la capacité de perception et de réflexion d'un individu, c'est-à-dire l'ensemble des données et des connaissances "internes" à un individu.
- La seconde est liée aux autres et à ce qui nous entoure : il s'agit de l'environnement externe (bruits, mouvements, etc), de la communication, de la structure organisationnelle du groupe et de la perception de l'activité en cours (i.e. activité mise en œuvre par les autres participants), c'est-à-dire l'ensemble des données et connaissances "externes" à un individu.

A partir de cette définition de l'activité de groupe qui résulte d'un ensemble d'actions dans un contexte donné [Karsenty 1997], nous définissons la structure de notre grille. En effet nous considérons quatre dimensions : les actions, l'observabilité des actions, les ressources du contexte supposé partagé et l'observabilité de ces ressources. Deux dimensions caractérisent les actions et leur contexte tandis que deux autres décrivent leurs observabilités respectives. Deux valeurs sont considérées sur chacune des quatre dimensions :



- Individuelle ou “interne” à un utilisateur
- Collective ou “externe”

En détaillant les quatre dimensions, nous montrons que cette structure nous permet d’atteindre notre objectif annoncé : intégrer les aspects du modèle du trèfle (spécifications fonctionnelles) et les propriétés ergonomiques (spécifications externes).

- **Actions** : Deux valeurs sont considérées sur cette dimension intitulée Actions.
 - * *Actions collectives* : un collectif doit fondamentalement mettre en œuvre et favoriser les actions collectives. Pour détailler les actions collectives, nous reprenons les trois classes du modèle du trèfle : actions de production, actions de communication et actions de coordination.
 - * *Actions individuelles* : les actions individuelles ne sont pas toujours possibles dans un collectif. Notons qu’un couplage fort de l’interaction (abordé dans la section intitulée “Spécifications externes et propriétés ergonomiques” du paragraphe 3.3) ne permet pas à un utilisateur d’effectuer une action individuelle. L’utilité d’autoriser des actions individuelles au sein d’un collectif dépend du domaine d’application. Néanmoins, de nombreuses études montrent qu’une activité de groupe n’est efficace que si le collectif aménage des sessions ou des espaces dédiés aux actions individuelles : un travail de groupe est le résultat d’un entrelacement d’actions collectives et d’actions individuelles.
- **Observabilité des actions** : selon cette dimension, nous avons identifié deux cas d’observabilité correspondant aux deux types d’actions de la dimension Actions. Nous considérons l’observabilité des actions :
 - * *Actions collectives* : une action collective qui participe directement à l’activité de groupe est caractérisée, par définition, par la propriété de rétroaction de groupe et par la propriété de couplage *WYSIWIS*.
 - * *Actions individuelles* : une action collective qui participe indirectement à l’activité de groupe est caractérisée, par définition, par la propriété de conscience de groupe.
- **Ressources du contexte supposé partagé** : il existe par définition deux types de ressources constituant le contexte des actions : les ressources collectives et les ressources individuelles. Nous retrouvons ici la notion de contextes externe et interne décrite dans [Karsenty 1997].

- **Observabilité des ressources du contexte supposé partagé** : les ressources du contexte sont essentielles pour interpréter les actions. Cette interprétation repose donc sur les ressources disponibles, c'est-à-dire observables. Pour caractériser l'observabilité de ces ressources, nous nous basons sur les propriétés suivantes : identification, réflexivité, réciprocité et filtrage. Nous identifions deux types de ressources observables ;
 - * *Ressources collectives* ; il s'agit des ressources propres à un groupe d'utilisateur ou communes à tous les utilisateurs : ces ressources collectives peuvent être rendues observables pour un groupe et ne pas l'être pour les autres.
 - * *Ressources individuelles* : il s'agit de ressources propres à chaque utilisateur rendues observables en les partageant. Entre autre, le filtrage définit le niveau de partage des ressources individuelles. En effet, toutes les ressources individuelles ne sont pas systématiquement rendues observables et une partie peut rester entièrement privée.

Les dimensions présentées à la Figure 13 constituent un cadre conceptuel pour décrire l'activité de groupe en intégrant des éléments de conception ergonomique. Cette grille d'analyse sert de fondement à nos travaux dont les objectifs sont : concevoir et élaborer des outils pour la conception et la réalisation logicielles qui soient complémentaires aux outils existants et centrés sur l'activité de groupe.

Comme nous l'avons exposé, les phases de réalisation logicielle s'organisent en deux étapes, la conception logicielle selon un modèle d'architecture conceptuelle puis la réalisation logicielle grâce à un outil de développement. Le chapitre suivant aborde la première étape en présentant un état de l'art des architectures conceptuelles pour les collecticiels selon notre grille d'analyse.

