

UNIVERSITE EL HADJ LAKHDAR - BATNA

Faculté des Sciences de l'Ingénieur
Département d'Informatique

THÈSE DE DOCTORAT

Spécialité : Informatique Industrielle

Présentée pour obtenir

LE GRADE DE DOCTEUR ES SCIENCES

Par

Ahmed KORICHI

Sujet de la Thèse

TCAO et Simulation : Vers une plate-forme
d'analyse et de conception de systèmes de
production orientée groupe

Soutenu le : 15 Janvier 2009

Devant le jury composé de :

Président	Pr. Nouredine Bouguechal	Professeur, université de Batna
Examineurs	Dr. Mohamed-Khireddine Krolladi	Maitre de conférences, université de Constantine
	Dr. Azzeddine Bilami	Maitre de conférences, université de Batna
	Dr. Allaoua Chaoui	Maitre de conférences, université de Constantine
	Dr. Okba Kazar	Maitre de conférences, université de Biskra
Rapporteur	Dr. Brahim Belattar	Maitre de conférences, université de Batna

Remerciements

Ecrire une thèse est un travail long et souvent difficile, aussi pour ceux qui m'ont aidé et accompagné ces dernières années. Finalement, ce mémoire de thèse est une œuvre collective puisque de nombreuses personnes y ont participé, de près ou de loin, avec leur soutien scientifique, moral ou affectif.

J'exprime mes profonds remerciements à mon directeur de thèse, le Docteur Brahim Belattar, Maître de Conférences au département d'Informatique, Faculté des Sciences de l'Ingénieur à l'université de Batna, qui a dirigé ce travail, pour son soutien, pour la sympathie qu'il m'a témoigné et pour la liberté de recherche qu'il a bien voulu me laisser.

Je remercie vivement Monsieur le Pr. Nouredine Bouguechal, Professeur à l'université de Batna, de m'avoir fait l'honneur d'être président de mon jury.

Je remercie très sincèrement le Dr. Mohamed-Khireddine Kholadi, maître de conférences à l'université de Constantine, d'avoir accepté de faire partie de mon jury de thèse.

Je remercie très sincèrement Monsieur le Dr Azzedine Bilami, maître de conférences à l'université de Constantine, d'avoir accepté de faire partie de mon jury de thèse.

Je remercie très sincèrement Monsieur le Dr Allaoua Chaoui, maître de conférences à l'université de Constantine, d'avoir accepté de faire partie de mon jury de thèse.

Je remercie très sincèrement Monsieur le Dr Okba Kazar, maître de conférences à l'université de Biskra, d'avoir accepté de faire partie de mon jury de thèse.

Je ne pourrais terminer sans remercier ma mère et ma famille ainsi que mon épouse qui m'ont soutenu et encouragé pour terminer ce travail. Que ma mère et mon épouse tout particulièrement trouvent ici ma reconnaissance.

Ahmed Korichi,

Table des matières

I	Introduction	1
1	Inéluçtabilité technologique.....	2
2	Sujet.....	2
3	Objectifs	3
4	Structure de la thèse.....	4
II	Le domaine de la simulation.....	6
1	Introduction.....	6
2	Les systèmes de production.....	7
3	La simulation.....	11
3.1	Notions de Système, Modèle et Modélisation	14
3.2	Domaine d'application de simulation	19
3.3	Les étapes d'une étude de simulation.....	21
3.4	Les outils de simulation	25
3.5	Les logiciels de simulation	27
4	Simulation et systèmes de production	31
4.1	Complexité du système de production	31
4.2	Limites des méthodes analytiques	32
4.3	Apports de la simulation aux développements du domaine de production.....	33
4.4	Champs d'application de la simulation aux systèmes productions	34
4.5	Exemples d'utilisation des logiciels de simulation dans le domaine de production.....	37
5	La conception coopérante d'un système de production.....	40
6	Entreprise-projet de simulation	43
6.1	Les acteurs d'un projet de simulation	45
6.2	Les moyens de coordination	48
6.3	La démarche	52
6.4	Durée du projet.....	52
7	État de l'art.....	53
7.1	Impact de la recherche sur le logiciel de simulation	53
7.2	Simulation et TCAO	54
8	Conclusion.....	59
III	Les Collecticiels	60
1	Introduction.....	60
2	Terminologies	61

3	Les fonctionnalités d'un collecticiel.....	64
4	Paramètres des collecticiels.....	67
4.1	Modes d'interaction : collecticiels synchrones/asynchrones.....	67
4.2	Le mode de fonctionnement WYSIWIS.....	68
4.3	Cohérence.....	69
4.4	Granularité.....	69
4.5	Droits d'accès, droit de parole, politiques à jetons.....	69
4.6	Accès libres, conflits d'accès et synchronisation.....	70
4.7	Gestion dynamique de groupes : protocoles de connexion/déconnexion de participants dans un collecticiel.....	71
4.8	Session.....	72
4.9	Les concepts IHM.....	72
5	Classifications des collecticiels.....	73
5.1	Systèmes pour la communication.....	73
5.2	Systèmes pour la production.....	75
5.3	Systèmes pour la coordination.....	76
5.4	Systèmes intégrés.....	77
6	Outils pour la mise en œuvre d'un collecticiel.....	80
6.1	Cycle de vie logiciel.....	80
6.2	Outils pour l'analyse des besoins.....	81
6.3	Outils pour les spécifications fonctionnelles et externes.....	86
6.4	Spécifications externes et propriétés ergonomiques.....	87
6.5	Outils pour la conception logicielle.....	90
6.6	Outils pour la réalisation logicielle.....	90
6.7	Outils pour l'évaluation ergonomique.....	92
7	Réalisations existantes.....	93
8	Conclusion.....	97
IV	Modèles conceptuels d'un collecticiel de simulation.....	98
1	Introduction.....	98
2	Hypothèses.....	99
3	Exigence pour la conception de collecticiel.....	100
3.1	La malléabilité.....	101
3.2	Multi-personnalisation.....	101
3.3	L'ouverture.....	102
3.4	L'articulation du travail.....	103
3.5	Intégration au Web.....	104

4	Les modes de coopération	104
5	Analyse des besoins.....	106
5.1	Modèle de tâche (GTA).....	107
5.2	Application de GTA au projet de simulation.....	112
6	Le modèle de Denver	121
6.1	Situation d'interaction	121
6.2	Protocole social de l'interaction.....	122
7	Spécifications fonctionnelles	124
7.1	Primitives de coopération.....	126
8	Modèle d'architecture logiciel.....	128
8.1	Production d'une architecture.....	128
8.2	Modèles de références.....	130
8.3	Génération d'un modèle d'architecture.....	145
9	Spécifications externes et propriétés ergonomiques	151
9.1	Critères ergonomiques	151
9.2	Métaphore de l'Interface utilisateur.....	154
9.3	L'interface et le modèle du trèfle.....	155
10	Conclusion.....	156
V	Implémentation	157
1	Introduction	157
2	Choix de collecticiel	157
2.1	Grille d'analyse d'un collecticiel.....	157
	Les services de communication.....	158
	Gestion de la coordination.....	159
	La conscience de groupe	160
	L'architecture système	160
	2.2 Le collecticiel retenu.....	161
3	Le BSCW.....	163
3.1	Mise en œuvre du système BSCW	164
3.2	Les fonctionnalités du système BSCW	166
3.3	Les Systèmes similaires	169
4	Expérimentation.....	170
4.1	Projet et cadre de l'expérimentation.....	170
4.2	Objectifs de l'expérimentation.....	171
4.3	Méthodes.....	171
4.4	Limites de l'expérimentation.....	173

4.5	Déploiement du projet de simulation	174
5	Résultats.....	178
6	Conclusion.....	180
VI	Conclusion.....	181
1	Bilan de l'étude réalisé.....	181
2	Apport.....	183
3	Limitations et extensions.....	183
4	Perspectives.....	184
VII	Références bibliographiques	185

Tables des Illustrations

Figure 1: Décomposition en systèmes d'une entreprise (Livet, 2002).....	7
Figure 2: Décomposition fonctionnelle (Livet, 2002).	9
Figure 3: Décomposition d'un système de production (Livet, 2002).....	10
Figure 4: Processus de modélisation (Belattar, 2000)	16
Figure 5 : Les étapes d'une étude simulation (Belattar, 2000).....	22
Figure 6 : Les différents niveaux d'utilisation (Hamadeh, 1993)	27
Figure 7 : le cheminement usuel de la simulation en utilisant.....	30
Figure 8 : Croissance du besoin en performance du système,	32
Figure 9 : Coût de changement effectué	35
Figure 10 : Différents modes d'organisation du travail coopératif. (Tarpin-Bernard, 1997b).....	42
Figure 11 : Les familles acteurs d'un projet de simulation	47
Figure 12 : Liste de participants d'un projet de simulation	48
Figure 13 : Trèfle du Collecticiel (Tarpin-Bernard, 1997b).....	64
Figure 14 : Cycle de vie en V de développement logiciel (Laurillau, 2002)	80
Figure 15 : Les cinq axes caractérisant la situation d'interaction (Salvador et al., 1996).....	84
Figure 16 : Les six axes caractérisant le protocole d'interaction (Salvador et al., 1996).....	85
Figure 17 : Construction de la structure de tâche dans l'outil EUTERPE.....	108
Figure 18 : Concepts et Relations dans GTA (Van der Veer et al., 1996).....	111
Figure 19 : Arbre des tâches	120
Figure 20 : Les situations d'interaction pour les quatre modes de coopérations	122
Figure 21 : Les protocoles d'interaction des quatre modes de coopérations	124
Figure 22 : Etapes de production d'une architecture (Laurillau, 2002)	129
Figure 23 : Le modèle centralisé (Salber, 1995).....	131
Figure 24 : Le modèle répliqué (Salber,1995)	132
Figure 25 : Le modèle hybride (Salber,1995)	134
Figure 26 : Différentes formes du modèle Zipper (Laurillau, 2002).....	135
Figure 27 : Modèle d'architecture de Dewan (Laurillau, 2002)	136
Figure 28 : Le modèle ALV (Salber,1995).....	137
Figure 29 : Les Agents du modèle ALV (Laurillau, 2002)	138
Figure 30 : Un agent PAC (Laurillau, 2002).....	139
Figure 31 : Modèle de référence Arch (Laurillau, 2002).....	140
Figure 32 : Modèle d'architecture PAC (Laurillau, 2002)	141
Figure 33 : Démentions fonctionnelle d'un Agent PAC (Laurillau, 2002).....	142

Figure 34 : Modèle d'Architecture Clover (Laurillau, 2002)	143
Figure 35 : Le Méta-Modèle d'Architecture Clover (Laurillau, 2002)	145
Figure 36 : Modèle Architecture logicielle pour un collecticiel de simulation	147
Figure 37 : Modèle de hiérarchie d'agents constituant la partie cliente	149
Figure 38 : Quatre niveaux d'observabilité (Laurillau, 2002).....	152
Figure 39 : Architecture du système BSCW (Bentley et al., 1996).....	165
Figure 40 : Environnement d'exécution du système BSCW (Bentley et al., 1996).....	166
Figure 41 : Espace du Chef de projet qui travail en mode Asynchrone	175
Figure 42 : Espace des étudiantes qui communiquent via X-LITE	176
Figure 43 : BSCW & NetMeeting en cours d'utilisation	177
Figure 44 : Changement du profil de communication du chef de projet	177
Figure 45 : Simulation du processus de remplissage des bouteilles de gaz.....	179
Figure 46 : Résultats de sondage.....	180
Tableau 1 : Méthode de construction et d'exploitation d'un modèle (Erard et Deguenon, 1999) ...	16
Tableau 2 : Les différents domaines d'applications de la simulation (GPA662, 2007)	20
Tableau 3 : Objectifs de la simulation (GPA662, 2007)	20
Tableau 4 : Liste de fonctionnalités requises dans un collecticiel.....	67
Tableau 5 : exemples type d'applications partagées à l'aide des systèmes intégrés	78
Tableau 6 : Les objets produits ou consommés et les actions associées	127
Tableau 7 : Rôles en fonction des données et primitives de coopération.....	128
Tableau 8 : Matrice des concepts observables.	153
Tableau 9 : Services de communication (Malcurat 2001).....	159
Tableau 10 : Fiche d'évaluation du collecticiel BSCW.....	163

Chapitre n° 1

Introduction

I Introduction

L'informatique dans son évolution historique prend progressivement en compte les souhaits des utilisateurs en s'adaptant à leurs exigences dans leur contexte de travail. C'est ainsi que, du travail par lot, on est passé au conversationnel, d'abord textuel, puis graphique et maintenant multimédia, voire multimodal c'est-à-dire intégrant plusieurs modes de communication (image, son, geste, etc.) compréhensibles par l'homme et la machine. On a également évolué de la vision individualiste des utilisateurs vers une approche plus collective du travail. L'ignorance mutuelle totale, entretenue par les machines virtuelles protégeant entièrement les utilisateurs, leurs données et leurs applications, a d'abord été rompue par le partage des données, puis par la possibilité de communiquer via des messageries. Maintenant, l'informatique aborde un nouveau champ d'investigation. Avec les progrès incessants des réseaux et télécommunications, il est aujourd'hui possible de faire travailler de façon coopérative plusieurs personnes distantes. C'est ce que l'on appelle le travail coopératif assisté par ordinateur ou TCAO.

Aujourd'hui, les entreprises et les particuliers sont de plus en plus nombreux à être équipés d'ordinateurs très performants munis d'accessoires variés (Webcam, Scanner, Ecouteurs, etc.). Par ailleurs, la démocratisation de l'accès à Internet, le nombre croissant de réseaux d'entreprises et de particuliers (i.e. Intranet et Internet) font que les utilisateurs s'orientent de plus en plus vers un mode de travail communautaire et géographiquement ouvert. Ces derniers deviennent de plus en plus exigeants, et attendent de l'outil informatique qu'il les assiste dans leur travail quotidien. On ne peut désormais plus envisager un programme informatique comme une application monoposte, utilisée par une personne isolée. La solution qui a consisté dans le passé à investir dans le développement d'applications typiquement centralisées et mono-utilisateur est en passe d'être écartée (Belattar et al., 1997). La prise en compte de la dimension de groupe dans la conception des applications est devenu un facteur prédominant. Certaines applications qui par le passé étaient proposées sous forme d'applications mono-utilisateur (Editeur de texte, Outil de C.A.O.) existent aujourd'hui sous forme d'application à l'usage d'un groupe (Groupware Application).

Dans ce contexte l'objectif de nos travaux est l'adaptation des résultats des recherches issus du domaine du travail coopératif assisté par ordinateur (TCAO) par vocation généraliste, à un domaine spécifique, la conduite d'un projet de simulation.

1 Inéluctabilité technologique

L'informatisation de la société est désormais inéluctable, qu'on le veuille ou non. Le réseau informatique Internet conquiert chaque jour une place plus importante chez les professionnels qui profitent de son faible coût d'exploitation, de sa facilité d'usage et de son universalité. Cette révolution emboîte le pas à une révolution plus ancienne, celle de l'informatisation des années 1980, et c'est tout logiquement que se trouvent renforcés les échanges et questionnés les protocoles de collaboration et les pratiques coopératives (Malcurat 2001).

Les logiciels de première génération, logiciels dits « métier » ne couvrant que les besoins d'une pratique professionnelle et généralement un seul aspect de celle-ci (dessins, calculs, estimations chiffrées, plannings), n'ont pas amené de modifications profondes dans l'organisation des pratiques professionnelles. Bien que l'aspect collaboratif n'y soit traité que de manière très limitée, notamment par les formats d'échanges de documents, ils ont cependant ouvert une perspective à la collaboration considérée comme un thème de réflexion et un champ de recherche à part entière.

L'outil informatique n'est pas innocent dans les transformations des pratiques professionnelles de coopération. La question des nouvelles technologies n'est pas un phénomène annexe qui ne relèverait que du champ des techniques et que l'on devrait traiter avec distance. Elle est au cœur de la question posée de la coopération non seulement parce qu'elle rend possible les échanges mais parce qu'elle les structure. Peut-être faut-il donner un exemple pour comprendre que les questions afférentes aux techniques de l'information ont des effets majeurs sur le système social tout entier. Le téléphone et le fax ont transformé les modes de collaboration et de coopération en accroissant considérablement les flux d'informations et en rendant les échanges plus synchrones. Ce raccourcissement du temps dans les transactions a modifié les comportements de tous. Dans le monde économique, il a contribué à remodeler les systèmes organisationnels qui sont de moins en moins construits selon des structures hiérarchiques et de plus en plus sur des structures en réseau.

2 Sujet

En Informatique, la tendance actuelle est plus que jamais à la décentralisation de l'informatique : c'est l'ère des stations de travail, des micro-ordinateurs, des portables et des ordinateurs reliés par un réseau informatique. La solution qui a consisté par le passé à investir des sommes importantes dans le développement d'applications typiquement centralisées et mono - utilisateur est en passe d'être écartée. La prise en compte de la notion de groupe dans la conception des applications est devenu un facteur prédominant, favorisée par les possibilités de connexion via un réseau informatique d'un

certain nombre d'individus devant coopérer pour la réalisation d'un projet commun mais aussi par la baisse des prix du matériel informatique et les offres de connexion gratuites à l'Internet. Certaines applications qui par le passé étaient proposées sous forme d'applications mono - utilisateur (Editeur de texte, Outil de C.A.O.) existent aujourd'hui sous forme d'application à l'usage d'un groupe (Groupware Application).

Dans le domaine de la simulation, Il est communément reconnu que pour pouvoir utiliser correctement et intelligemment des méthodes de simulation, il faut disposer de connaissances plus ou moins solides dans des domaines variés (Probabilités et Statistiques, Modélisation, Programmation, etc.). Malheureusement, il est souvent rare pour ne pas dire impossible de trouver au sein de l'entreprise un cadre possédant à lui seul toutes les connaissances requises pour pouvoir mener à terme un projet de simulation sans avoir recours aux connaissances d'un certain nombre d'individus internes ou externes à son entreprise. Par conséquent, un projet de simulation quelle que soit son envergure, ne peut et ne pourra jamais être considéré comme l'œuvre d'une personne isolée et ce même si les attributions de rôle dans le projet ne sont pas faites de manière explicite.

Partant de ce constat, on explore les possibilités d'intégration des concepts du travail coopératif assisté par ordinateur dans un environnement de modélisation et de simulation en vue de dégager l'architecture d'une nouvelle génération d'outils basés essentiellement sur le travail de groupe.

3 Objectifs

Les objectifs de notre étude sont la spécification d'un environnement logiciel pour assister le travail coopératif dans la conduite d'un projet de simulation. Nous adoptons un regard nouveau sur les outils de simulation existants, en étudiant comment de tels outils peuvent intégrer les concepts du TCAO et des aspects de la conception comme ceux relevant de l'ergonomie ou de l'activité de groupe. Ainsi nous visons la mise en œuvre d'un collecticiel de simulation guidée non seulement par des aspects technique mais aussi en tenant compte des facteurs sociaux, éthiques et ergonomiques. Cette visions nous conduit à explorer des domaines assez varies afin d'établir des analogies avec le domaine de simulation.

Vis-à-vis de nos objectifs, nous identifions les étapes de travail suivantes :

- Analyse des pratiques de la simulation : Cette étape répond à deux objectifs : tout d'abord, face à l'étendue du problème posé, elle recadre précisément les domaines de notre recherche en se basant sur ce que nous pensons avoir un intérêt à traiter. Le second objectif consiste à donner des éléments d'analyse du contexte spécifique à notre étude ; le domaine de la conduite d'un projet de simulation. Les caractéristiques principales de ce domaine y sont développées, au regard de ce qui constitue la finalité de notre travail ; la spécification d'un collecticiel de simulation. Ainsi, un état de l'art des travaux portant sur la coopération et la collaboration dans le domaine de la simulation va nous permettre de mieux situer nos choix privilégiés.
- Etude des collecticiels : Pour mieux maîtriser les méthodes de conception et de réalisation des collecticiels, et en vue de les appliquer dans la phase de conception de notre collecticiel futur, cette étape sera consacrée essentiellement aux collecticiels. Cette étape va nous permettre de mieux cerner le domaine des collecticiels pour atteindre notre objectif qui est la conception et la réalisation d'un collecticiel de modélisation et de simulation. Dans un premier temps, nous maîtrisons les concepts liés aux collecticiels. Ensuite, nous établissons un état de l'art des outils de conception et de développement logiciel des applications de groupe en vue d'identifier celles qui conviendraient au mieux compte tenu des objectifs de ce travail de recherche.
- Etude conceptuelle : Sur la base des deux étapes précédentes et au regard du contexte spécifique au domaine de simulation étudié, dans cette étape, nous proposons une conception d'un collecticiel de simulation en utilisant les méthodes de conception retenue. Le domaine d'application à privilégier sera certainement l'analyse et la conception en groupe des systèmes de production.
- Implémentation & étude expérimentale : Dans cette étape nous proposons une implémentation d'un collecticiel de simulation sur la base des résultats des étapes précédentes. Ensuite, nous conduisons une série d'expérimentation dans le but de valider les modèles conceptuels proposés dans l'étape précédente.

4 Structure de la thèse

La structure en chapitres de cette thèse reflète notre démarche de recherche : Le deuxième chapitre cernerá précisément le contexte de collecticiel visé en mettant l'accent sur l'aspect coopératif de la méthode de conduite d'un projet de simulation. Au regard de ce qui constitue la base d'une spécification d'un collecticiel de simulation, les principaux éléments du domaine de la

simulation y sont développées. Dans la dernière partie de ce chapitre, nous présentons les travaux de recherches relatifs à la coopération et à la collaboration dans le domaine de la simulation.

Le troisième chapitre sera consacré essentiellement aux collecticiels. Nous définissons ces applications et nous identifions leurs objectifs, leurs caractéristiques et les réalisations qui ont été faites. Nous étudions ensuite les outils existants de conception et de développement des collecticiels afin de mieux maîtriser la mise en œuvre des applications collaboratives.

Le quatrième chapitre sera consacré à l'étude conceptuelle d'un collecticiel de modélisation et de simulation. Dans ce chapitre, nous proposons une conception d'un collecticiel de simulation guidée par des aspects technique et prend en compte les facteurs sociaux, éthiques et ergonomiques. Cette conception comporte essentiellement : un modèle d'analyse des besoins, un modèle architecture logicielle et un modèle d'interface.

Dans le cinquième chapitre, nous illustrons notre expérience avec le système BSCW pour implémenter un collecticiel de simulation orienté vers l'analyse et la conception en groupe des systèmes de production dans le but de valider les modèles élaborés.

Dans le dernier chapitre, nous concluons cette étude par une synthèse de notre contribution et nous montrerons à quel point notre objectif a été atteint. Nous exposerons ensuite les perspectives et les futurs travaux envisageables pour approfondir cette étude.

Chapitre n° 2

Domaine de Simulation

II Le domaine de la simulation

1 Introduction

La complexité de mise en place des systèmes de production exige le recours à une aide informatisée que ce soit pendant la phase de la conception et de la mise en place d'installations de production ou pendant le fonctionnement du système (Hamadeh, 1993). Les méthodes analytiques, trop restrictives, peuvent apporter une aide lors d'une évaluation approchée des performances du système envisagé. Par contre, les outils de simulation peuvent apporter aux différents intervenants d'un projet et selon les critères et les besoins de chacun, une connaissance sur le comportement du système étudié. Si la simulation est une aide à la conception, c'est également une aide à l'exploitation.

Dans le domaine de simulation il est communément reconnu que la réalisation d'un projet de simulation fait intervenir de nombreuses personnes avec des compétences et des rôles différents (Belattar et al., 1997). Jusqu'à une époque récente, les outils informatiques ne prenaient pas en compte cette dimension et laissaient au chef de projet le soin de gérer le processus de conception et sa dimension multi-participant hors système informatique.

L'objectif de ce chapitre est de déterminer les éléments d'analyse du domaine de la simulation en insistant sur les éléments qui peuvent constituer un point de départ pour la spécification d'une plate-forme d'analyse et de conception de système de production orientée groupe.

Le chapitre est organisé de la façon suivante : la première partie présente la simulation des systèmes de production, la méthode de conduite d'un projet de simulation et justifie l'adéquation des concepts du TCAO avec cette méthode. Dans la deuxième partie, nous présentons un bilan des recherches qui ont prédominé dans ce domaine ainsi que les travaux portant sur la coopération et la collaboration dans un projet de simulation.

2 Les systèmes de production

Tout système de production est une organisation dont la fonction est d'offrir des biens ou des services (Livet, 2002). Il se voit assigné des objectifs, physiques, monétaires ou sociaux, et ses résultats sont mesurés à l'aide d'indicateurs de performance à partir desquels sont menés des actions comme la re-conception, l'amélioration, l'optimisation, etc. Les concepteurs, les exploitants du système de production attendent donc que les processus (physiques et informationnels), les ressources (équipements, matières, hommes, ...etc.), les acteurs qui composent le système de production, concourent aux multiples objectifs assignés. La production est donc considérée comme la nouvelle arme de la compétitivité, et les entreprises industrielles se retrouvent de ce fait dans un environnement totalement modifié.

Pour identifier le système de production, il est à la fois nécessaire de définir sa place par rapport aux différentes fonctions de l'entreprise mais aussi d'identifier sa décomposition. En effet, le système de production peut être défini comme étant un système dont les activités sont intégrées entre elles mais aussi aux activités de l'entreprise.

Une entreprise industrielle, selon une approche système, se décompose en divers systèmes tels que la production, l'administratif, le commercial, eux-mêmes composés de nombreuses fonctions. Ces différentes parties forment un réseau étroitement lié par des flux d'informations, de matières, de décisions etc. comme le montre la Figure suivante.

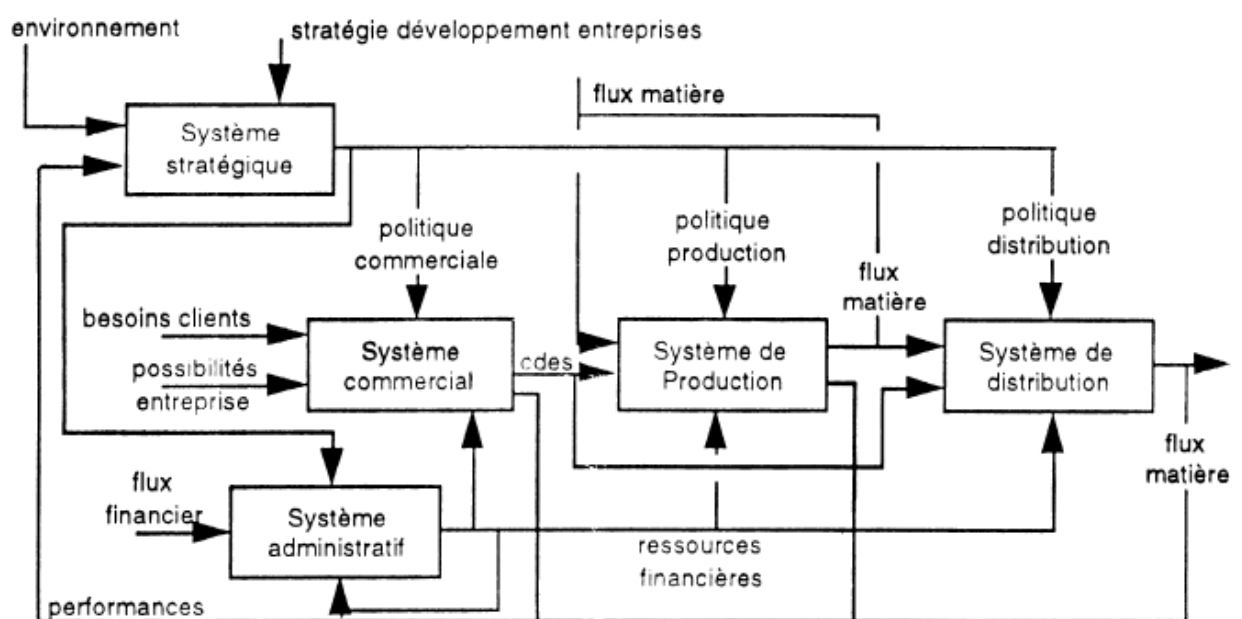


Figure 1: Décomposition en systèmes d'une entreprise (Livet, 2002).

D'autre mode de décomposition de l'entreprise propose une décomposition fonctionnelle (Livet, 2002). Selon l'auteur de cette décomposition, pour comprendre la façon dont les organisations se structurent, il faut en expliquer le fonctionnement, c'est-à-dire identifier :

- Les éléments qui en font partie,
- Les liens qui unissent ces éléments,

Les éléments de base de l'organisation d'une entreprise sont au nombre de cinq (Figure 2) :

- **Le centre opérationnel** : Il rassemble les membres de l'organisation (opérateurs) dont le travail est directement lié à la production des biens et des services. Le centre opérationnel procure les entrées nécessaires à la production, fabrique et distribue les produits et assure la maintenance.
- **Le sommet stratégique** : Sa fonction est de faire en sorte que l'organisation remplisse sa mission de façon efficace et qu'elle serve les besoins de ceux qui la contrôlent ou qui ont sur elle du pouvoir (propriétaires, administrations etc.). Il exerce des fonctions de supervision directe : allocation des ressources, règlement des conflits, contrôle, diffusion de l'information, leadership. Il est également chargé de la gestion des conditions de travail, de l'organisation et de ses relations avec l'environnement (contacts à haut niveau, négociations etc.). Enfin, il doit développer la stratégie de l'organisation.
- **La ligne hiérarchique** : C'est la ligne d'autorité composée de cadres, contremaîtres etc., qui relie le sommet stratégique au centre opérationnel. Chaque membre de la chaîne hiérarchique accomplit, à son niveau, le travail du sommet hiérarchique.
- **Technostructure** : Elle est composée d'analystes chargés de la conception et de l'adaptation de la structure qui agissent sur le flux de travail par le biais de la standardisation : les hommes des méthodes standardisent le travail, les hommes de la planification et du contrôle standardisent les résultats et les hommes du personnel standardisent les qualifications.

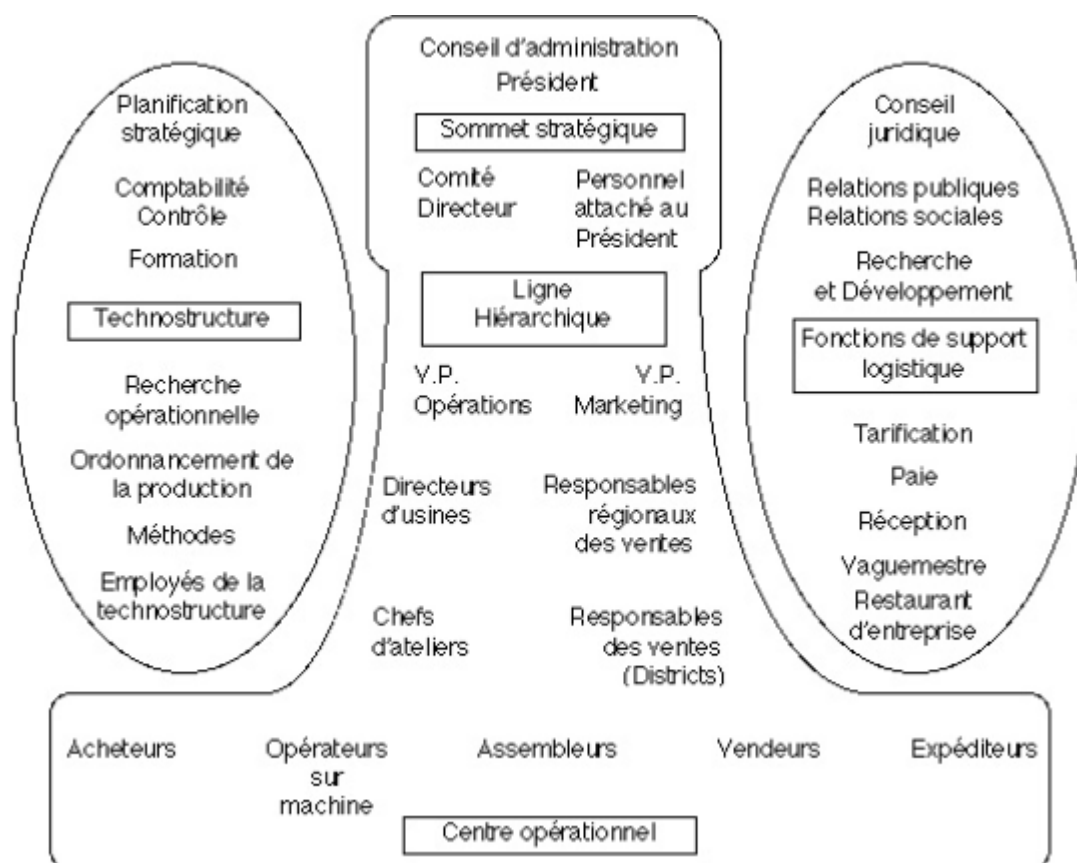


Figure 2: Décomposition fonctionnelle (Livet, 2002).

- o **Les unités fonctionnelles logistiques** : Il s'agit d'unités spécialisées qui ont une fonction particulière à remplir : recherche, contentieux, relations publiques etc. Elles interviennent indirectement dans le flux du travail.

Les éléments de cette décomposition élaborent un réseau étroit de flux d'informations, de matières et de décisions. Sa représentation de l'organisation (Figure 2) montre le centre opérationnel englobant la fonction de production qui n'en est pas moins une fonction névralgique de l'entreprise comme le montre la Figure 1.

Le système de production peut être à son tour morcelé. La théorie du système général propose plusieurs décompositions selon 9 niveaux dépendants du degré cognitif du système étudié. Le système de production selon cette théorie peut s'assimiler à « un objet actif qui se coordonne » correspondant au niveau 7 (Livet, 2002). A ce niveau, le système se décompose en 3 sous systèmes en interaction : le système de gestion de production, le système informationnel, le système physique de production.

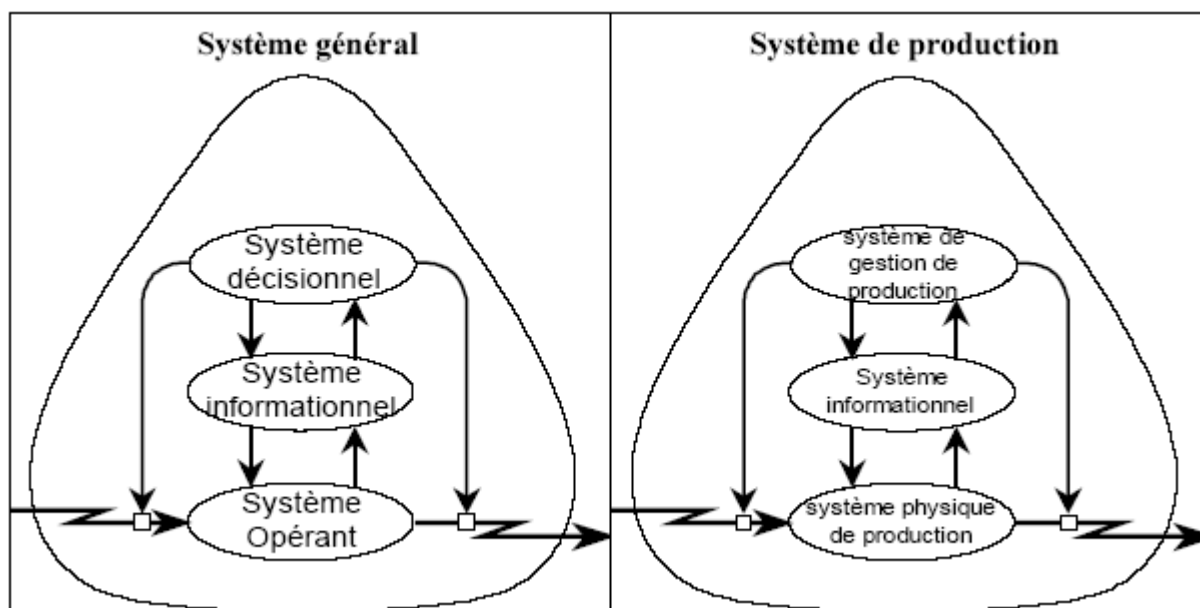


Figure 3: Décomposition d'un système de production (Livet, 2002).

- **Le système décisionnel** : Il prend les décisions nécessaires pour atteindre des objectifs techniques, économiques, sociaux et humains du système de production en s'appuyant sur les informations fournies par le système d'information.
- **Le système d'information** : Il regroupe l'ensemble des informations circulant ou stockées dans le système de production. Il assure la liaison entre les sous-systèmes du système de production.
- **Le système opérationnel** : Il représente le système concret de production. Il transforme les matières premières et les produits semi-finis.

Le contexte économique dans lequel les entreprises évoluent aujourd'hui ne permet plus de produire efficacement à partir des systèmes de production basés sur les principes du Taylorisme, qui ont fait largement leurs preuves. Il nécessite des systèmes de production basés sur d'autres principes, ayant des nouvelles caractéristiques, telles que flexibilité, réactivité et agilité. Ces caractéristiques sont des qualités incontournables pour des systèmes de production qui se trouvent confrontées à une demande variée et fluctuante avec des contraintes de qualité et de délais de plus en plus fortes. Le défi pour les entreprises est donc d'installer des outils de production modulaires et flexibles avec des systèmes de pilotage capables de les gérer.

D'autre part, les compétitions internationales auxquelles se livrent les industriels les conduisent à mettre en œuvre des systèmes automatisés de plus en plus onéreux et complexes pour lesquels les

méthodes traditionnelles de conception sont limitées à cause de leur incapacité à prendre en compte l'ensemble des composantes d'un système de production intégré. Par ailleurs, cette phase de conception initiale repose sur des hypothèses de production (types et quantités de produits à fabriquer) qui, avec le temps, peuvent ne plus correspondre à la réalité des commandes à satisfaire. La flexibilité des systèmes de production, c'est-à-dire leur capacité à s'adapter à une demande fluctuante, exige que leur conception soit remise en cause périodiquement. Cependant, il est communément reconnu que la simulation trouve tout son intérêt quand il s'agit d'analyser ou de concevoir des systèmes de production complexes (conception, re-conception, dimensionnement de système de production, ou comme un outil ponctuel d'aide à la décision), et elle est le seul outil disponible aujourd'hui dans la mesure où les méthodes analytiques traditionnelles imposent des simplifications abusives.

3 La simulation

La simulation est largement reconnue comme la technique la plus efficace pour l'analyse et la conception à la disposition des concepteurs et des gestionnaires des systèmes complexes. Elle peut être appliquée dans divers domaines, tels que, l'analyse des systèmes de services, les systèmes de production, les systèmes naturels, les systèmes informatiques. ..Etc. Cette technique connaît actuellement un essor considérable. Ceci est dû aussi bien à l'intérêt théorique que présente la modélisation des systèmes simulés, que par le besoin croissant de simuler par ordinateur des réalisations de plus en plus complexes.

La simulation utilise souvent des programmes informatiques pour imiter les événements causaux et les actions conséquentes dans le système. Les statistiques sont accumulées durant la période de simulation pour évaluer les mesures d'intérêts qui sont résumées et rapportées à la fin de l'exécution du programme de simulation. De ce fait, on peut dire que la simulation implique la génération d'un historique artificielle du comportement du système (Belattar, 2000), et l'observation d'un historique pareil pour connaître les caractéristiques opératoires du système réel.

Plusieurs personnes croient que la simulation par elle-même résout les problèmes. Ils supposent qu'en définissant le problème, la simulation générera une solution. De telles techniques sont retrouvées dans les systèmes experts et consistent en des règles et des heuristiques définies par un expert pour proposer des solutions à un ensemble donné de problèmes.

Les systèmes experts sont essentiellement des boîtes noires puisque l'utilisateur n'a pas besoin de connaître, et est souvent non intéressé à connaître, comment la solution est dérivée. Par contre, la simulation permet d'évaluer et non pas de générer des solutions. Elle ne produit pas de solutions théoriques optimales, mais plutôt elle oriente la recherche dans la direction de la meilleure solution pratique.

La simulation est utilisée pour investir une large variété de questions du type "Quoi Si" à propos d'un système réel. Les changements majeurs d'un système peuvent être tout d'abord simulés pour prédire leur impact sur la performance de celui-ci. La simulation peut être aussi utilisée pour étudier un système durant le stade de conception, et avant qu'il soit construit. Ainsi, la simulation peut être seulement utilisée comme outil d'analyse pour prédire l'effet des changements sur la performance d'un système existant, mais aussi comme un outil de conception pour prédire la performance d'un nouveau système sous différentes circonstances.

Pour atteindre les buts de la simulation, on devra généralement exprimer les propriétés importantes du système de manière à permettre de faire des expériences, c'est-à-dire de manipuler, varier les attributs de certaines composantes et la nature des interactions entre ces composantes. Le système sera donc exprimé sous forme de symboles, qui représentent les composantes, et d'expressions mathématiques ou logiques, qui représentent les relations fonctionnelles ou interactions; un tel processus est appelé abstraction, et l'ensemble des expressions symboliques est appelé un modèle du système réel. Et lorsqu'on parle de la réalisation d'une expérience de simulation, on parle de l'observation de la performance du modèle dans des conditions définies.

Au début des années quarante, les recherches militaires dans ce domaine sont orientées vers le développement des procédures formelles et efficaces pour aider à prendre des décisions concernant le design des systèmes de défense contre les attaques aériennes; les méthodes de recherche opérationnelle qui en ont résulté ont été améliorées de beaucoup depuis, et leurs champs d'application ont été agrandis, de sorte qu'elles sont utilisées aujourd'hui avec succès pour résoudre une grande variété de problèmes (GPA662, 2007). Ces méthodes utilisent des modèles symboliques et des processus mathématiques de déduction pour prédire les effets de différentes solutions proposées pour un problème donné et pour déterminer la solution optimale.

La recherche opérationnelle a de beaucoup amélioré le processus de prise de décision du gestionnaire situé à un niveau intermédiaire dans l'organigramme de l'entreprise; à ce niveau, en effet, la décision doit être prise en respectant des politiques et contraintes préétablies, sur lesquelles le gestionnaire n'a habituellement aucun contrôle. C'est à ce niveau que les méthodes de recherche

opérationnelle ont été utilisées pour le contrôle d'inventaire, l'analyse des systèmes de production. Malheureusement, lorsqu'on a tenté d'appliquer les méthodes de recherche opérationnelle aux prises de décision effectuées au niveau supérieur de la direction de l'entreprise, pour des problèmes tels que l'investissement de capital ou la planification de la gamme de produits, on s'est rendu compte que ces techniques mathématiques élégantes n'étaient plus aussi efficaces (GPA662, 2007). L'une des raisons de cette difficulté est qu'à ce niveau d'autorité, il y a moins de contraintes, ou encore ces contraintes sont moins fortes; par ailleurs, les valeurs de certains paramètres ne sont connues qu'avec une faible précision. On peut dire que le besoin de techniques quantitatives utilisables dans des situations complexes, de même que la disponibilité de plus en plus grande d'ordinateurs puissants, a amené le développement des méthodes de simulation.

Les méthodes modernes de simulation ont par ailleurs bénéficié du travail effectué par Van Neumann, Vlan et Fermi lors de leurs travaux sur le développement de la bombe atomique pendant la deuxième guerre mondiale (GPA662, 2007). Ces chercheurs ont en effet simulé le comportement d'un écran anti-rayonnements à l'aide de variables stochastiques et de la génération de nombres aléatoires pour donner une valeur à ces variables, et ceci pour pallier à la difficulté de résolution d'une expression mathématique décrivant un tel système. Cette technique est depuis connue sous le nom de Monte-Carlo.

Dans certaines situations, un modèle peut être développé de telle sorte qu'il est assez simple pour être résolu par les méthodes mathématiques. De telles solutions peuvent être générées en utilisant les calculs différentiels, la théorie des probabilités, les méthodes algébriques ou les techniques mathématiques. La solution consiste souvent en un ou quelques paramètres numériques qui sont appelés mesures de performance du système. Cependant, plusieurs systèmes réels sont assez complexes que les modèles correspondants sont impossibles à résoudre mathématiquement. Dans ce cas, la simulation peut être utilisée pour imiter le comportement du système. À partir de la simulation, les données sont collectées comme si elles sont en observant le système réel. Ces données sont alors utilisées pour estimer les mesures de performance du système.

En conclusion, la simulation sur ordinateur est un outil de prise de décision relativement récent, qui est déjà utilisé dans l'industrie. L'utilisation de cette technique pour l'analyse ou la conception d'un système de production ou de l'un de ses sous-systèmes comprendra la conception d'un modèle et l'étude du fonctionnement de ce modèle sous différentes conditions, afin de déterminer les conditions de fonctionnement favorisant l'atteinte des objectifs du système.

3.1 Notions de Système, Modèle et Modélisation

Les trois notions de système, Modèle et Simulation sont intrinsèquement liées. Nous allons définir de manière plus ou moins précise chacune d'elles avant de présenter les concepts liés à la méthode de conduite d'un projet de simulation.

Notion de système

Le mot "système" couvre un champ d'application immense et de nombreuses définitions lui ont été attribuées dans la littérature. Nous préférons celle donnée par l'AFCEI (Belattar, 2000) et qui s'énonce comme suit :

«Un système est une entité complexe traitée (eu égard à certaines finalités) comme une totalité organisée, formée d'éléments et de relations entre ceux-ci, les uns et les autres étant définis en fonction de la place qu'ils occupent dans cette totalité et cela de telle sorte que son identité soit maintenue face à certaines évolutions.»

Comme exemples de systèmes nous pouvons citer les exemples suivants :

- Un système biologique, comme le système circulatoire d'un mammifère, comporte divers organes (cœur, poumons, reins, tissus divers...) qui sont les entités de traitement du système, un organe de stockage (la rate), des veines et des artères (les moyens de communication du système). Les entités traitées par ce système sont principalement le sang (entité interne) et l'air (entité externe).
- Un système de production, comme un atelier de fabrication de pièces mécaniques, comporte des machines-outils (les entités de traitement), des transporteurs permettant d'acheminer matières premières et pièces en cours de fabrication, des stocks internes permettant d'entreposer les pièces en attente d'un usinage...
- Une raffinerie de pétrole contient divers éléments de stockage (matières premières à raffiner, produits en cours de raffinage ou finis), de traitement (tour de craquage par exemple) et de transport (conduits divers, pompes...). Un bureau de poste comporte divers guichets ; à chacun d'entre eux sont affectés un postier, des automates permettant le pesage et l'affranchissement (ce sont des organes de traitement), devant lesquels se forment des files d'attente (organes de stockage) où patientent des clients en attente d'un service.

- Un système écologique comporte, par exemple, un milieu herbeux (matière première) nourrissant une population de lièvres (les clients) dont se nourrissent à leur tour des lynx (autres clients). Lièvres et lynx se reproduisent, augmentant ainsi leur population, mais ils peuvent aussi contracter des maladies ou subir une famine, circonstances produisant l'effet inverse. Un tel système est nettement plus compliqué à appréhender que les précédents.

La simulation d'un système réel devient utile dès lors que la réalisation d'un modèle mathématique de ce système est, soit trop coûteuse en temps de conception ou de calcul, soit trop simplificatrice de la réalité, de sorte que le modèle obtenu ne permet plus d'appréhender les informations utiles sur le système à étudier.

Modèle et Modélisation

La modélisation consiste à construire une représentation simplifiée d'un système appelée généralement « modèle ». L'AFCEI (Belattar, 2000) définit un modèle par l'énoncé suivant :

«Un modèle est un schéma, i.e. une description mentale (intériorisée), ou figurée (diagrammes, formules mathématiques, etc ... qui, pour un champ de questions est pris comme représentation abstraite d'une classe de phénomènes, plus ou moins habilement dégagés de leur contexte par un observateur pour servir de support à l'investigation, et/ou la communication».

Un modèle est donc une représentation d'un système (réel ou imaginaire) dont le but est d'expliquer et de prédire certains aspects du comportement de ce système. Cette représentation est plus ou moins fidèle car d'une part, le modèle devra être assez complet afin de pouvoir répondre aux diverses questions qu'on peut se poser sur le système qu'il représente, et d'autre part, il ne doit pas être trop complexe pour pouvoir être facilement manipulé. Ceci implique qu'il y a un intérêt à bien définir les limites ou les frontières du modèle qui est censé représenter le système.

Il existe différents types de modèles. Les modèles physiques sont ceux dans lesquels le système réel est représenté par une réplique ou une maquette, à une échelle différente, et éventuellement à l'aide de matériaux différents (exemple : maquette de véhicules pour les essais aérodynamiques en soufflerie). Les modèles symboliques sont une abstraction mathématisée de la réalité. Ils sont généralement exécutés sur un ordinateur. Une autre distinction concerne la prise en compte des aléas dans le modèle. Dans certains cas, qualifiés de déterministes, leur influence est considérée comme négligeable. Le plus souvent, ils doivent être représentés car ils jouent un rôle significatif

(exemple typique : les pannes). On a alors affaire à des modèles stochastiques. Une troisième dichotomie sépare les modèles statiques et les modèles dynamiques. Dans les premiers, le temps n'intervient pas (exemple : modèle comptable permettant de calculer un profit en fin d'exercice à l'aide d'un tableur). En revanche, dans les seconds, il est un facteur essentiel du comportement et de l'état du système (exemple : réacteur chimique régi par des équations différentielles). Enfin, à l'intérieur des modèles dynamiques, on distingue les modèles discrets, dans lesquels l'état du système ne change qu'à certaines dates (exemple : une file d'attente devant un guichet), et les modèles continus ou ce changement est permanent (cas du réacteur déjà cité). Un modèle qui contient à la fois des composantes discrètes et continues est dit mixte.

Dans cette étude nous nous sommes intéressés seulement à la simulation sur ordinateur c'est-à-dire les modèles symboliques exécutés sur ordinateur. La méthode de construction et d'exploitation du modèle dépend de la nature du système réel et de sa représentation (Tableau 1) :

Systeme	Modèle	Méthode
Déterministe	Déterministe	Analyse numérique
Stochastique	Déterministe	Calcul des probabilités
Déterministe	Stochastique	Monte Carlo
Stochastique	Stochastique	Simulation

Tableau 1 : Méthode de construction et d'exploitation d'un modèle (Erard et Deguenon, 1999)

Le processus de construction d'un modèle de simulation peut être schématisé comme suit :

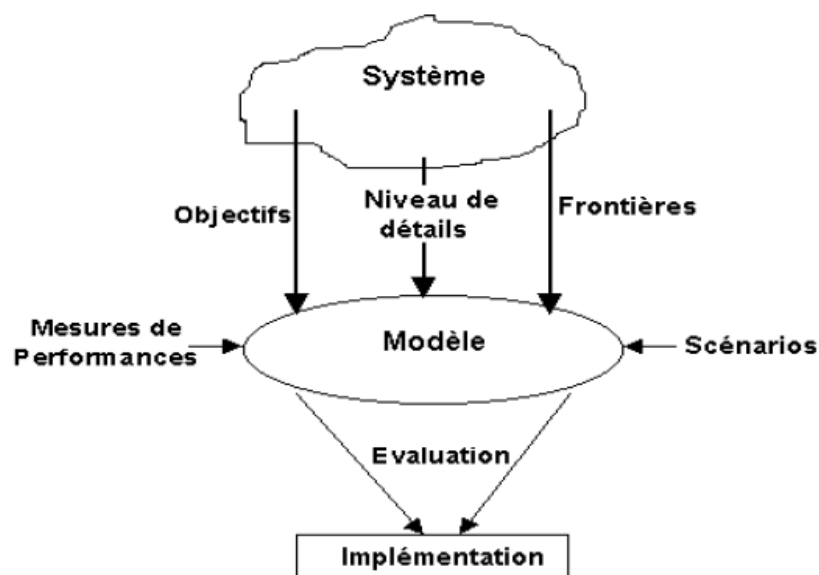


Figure 4 : Processus de modélisation (Belattar, 2000)

Le processus de modélisation permet de préciser le contexte de l'étude. Il comprend l'identification du problème; la spécification des objectifs, et la réalisation d'une première ébauche du modèle qui a pour but d'en délimiter les frontières et de spécifier les données dont on a besoin et la validation de ce modèle auprès de l'utilisateur.

Le but à atteindre par ce processus est de construire un modèle valide tout en restant cohérent avec les objectifs de l'étude. Il faut donc tout d'abord formuler explicitement ces objectifs et les divers scénarios à étudier. A ce niveau, il faut souligner que le concepteur se trouve devant un compromis difficile à faire. En effet, le concepteur du modèle cherche toujours la simplification, alors que le client souhaite que soient finement représentés les constituants du système. Le processus de construction d'un modèle comprend la modélisation logico-mathématique qui peut être facilitée par un outil graphique, et la programmation proprement dite. Donc ce processus conduit à la construction de deux modèles : Le modèle conceptuel qui est la représentation logico-mathématique du système réel. Il est obtenu dans une phase d'analyse et de modélisation. Le modèle programmé qui est la mise en œuvre du modèle conceptuel sur un ordinateur. Il est obtenu dans une phase de programmation et de mise en œuvre. Quand ce dernier est validé, il peut servir à l'évaluation du comportement dynamique du système. Cette évaluation nécessite une définition précise de la campagne d'exploitation, la production des mesures par la simulation proprement dite, la mise en forme et la comparaison des résultats obtenus aux objectifs poursuivis. S'ils n'ont pas été atteints, de nouveaux scénarios sont proposés et testés jusqu'à satisfaction.

Simulation

Etymologiquement, le terme "simulation" est dérivé du mot latin "SIMULARE" qui veut dire : copier, feindre, faire paraître comme réelle une chose qui ne l'est point. On peut donc très bien dire que simuler le fonctionnement d'un système c'est imiter son fonctionnement au cours du temps en manipulant un modèle. Ceci équivaldrait à la génération d'un historique artificiel des changements d'état du système et l'observation de cet historique pour faire des déductions sur ses caractéristiques de fonctionnement. C'est donc une méthodologie essentiellement pratique qui permet de modéliser aussi bien des systèmes conceptuels que des systèmes existants déjà. La simulation peut être utilisée pour décrire et analyser la dynamique d'un système, répondre aux questions de type «Quoi si ? » sur le système réel et aider à la conception d'un système réel.

Dans la littérature technique beaucoup de définitions ont été attribuées au terme “Simulation”. Parmi ces définitions, nous préférons celle donnée par A.A.B. Pritsker (Belattar, 2000), et qui s'énonce comme suit :

«La simulation est l'étude du comportement dynamique d'un système, grâce à un modèle que l'on fait évoluer dans le temps en fonction de règles bien définies, à des fins de prédiction».

Toute étude de simulation suppose que l'on ait identifié clairement les caractéristiques du système qui sont nécessaires à la détermination des performances recherchées. La simulation est très utile dans les cas suivants :

- Le système n'est pas décomposable en sous-systèmes plus simples et indépendants les uns des autres. C'est la condition essentielle qui impose une simulation de ce système. Au contraire, un système décomposable sera appréhendé après décomposition en ses éléments constitutifs. C'est ainsi que la biologie décompose traditionnellement le système complexe que constitue l'organisme d'un mammifère en sous-systèmes qui sont considérés comme indépendants, parmi eux : le système circulatoire...
- Le système n'existe pas encore et une étude préliminaire (phase de conception) est en cours. Par exemple, pour préciser le dimensionnement d'un atelier de fabrication de pièces mécaniques, il faut évaluer les performances que doivent réaliser les machines à utiliser, leur nombre, leur emplacement, les transporteurs nécessaires, puis il faut vérifier le fonctionnement correct du système en projet, son adéquation aux performances attendues...
- Les expériences sur le système réel coûtent cher, ou sont impossibles à réaliser pendant son fonctionnement. Par exemple, modifier le rythme cardiaque dans le système circulatoire pour en mesurer l'impact est une expérience peu recommandée.
- Le système est inaccessible : le système circulatoire en est un excellent exemple.
- Les temps d'observation nécessaires sur le système réel sont incompatibles avec les besoins : l'évolution du système est très rapide, de sorte que, à l'échelle humaine, les changements d'état du système semblent continus, ou cette évolution est au contraire très lente, de sorte que le système n'évolue pas à l'échelle de temps dont on dispose pour

mener l'étude (c'est généralement le cas d'un système écologique, qui évolue très lentement à l'échelle humaine).

- Faire des expériences sur le système réel est dangereux ou peut avoir des conséquences graves : pensons au système circulatoire, ou au danger que peut représenter la mise en œuvre d'une expérimentation dans une raffinerie de pétrole... Rappelons que la catastrophe de Tchernobyl s'est produite au cours d'un entraînement de sécurité pendant lequel est survenu un véritable incident... Si cet entraînement avait été réalisé sur un modèle de simulation les conséquences auraient été moins dramatiques

3.2 Domaine d'application de simulation

La simulation a été, pour la première fois, utilisée en 1950 dans la planification stratégique au niveau militaire. Elle n'a gagné sa popularité dans les domaines manufacturiers et de services qu'au début des années quatre-vingt. Pour plusieurs compagnies, la simulation est devenue une pratique standard lorsqu'une nouvelle usine est à implémenter ou un changement de procédé nécessite d'être évalué. Les études indiquent que la simulation figurait parmi les techniques les plus utilisées à travers le monde (Belattar, 2000).

Plusieurs facteurs ont contribué à l'augmentation de l'utilisation de la simulation dont on cite :

- Croissance de la conscience et de la compréhension des nouvelles technologies (automatisation).
- Augmentation de la disponibilité, de la capacité et de la facilité d'utilisation des logiciels de simulation.
- Augmentation des capacités en mémoire et en vitesse de traitement des ordinateurs (surtout au niveau des PC).
- Chute des coûts des ordinateurs.
- Adoption répandue des micro-ordinateurs.
- Disponibilité de l'animation graphique.

La disponibilité de logiciels de simulation spécialisés et faciles à utiliser et d'ordinateurs puissants a non seulement rendu la simulation plus acceptable, mais aussi plus accessible aux concepteurs et aux gestionnaires qui n'ont ni le temps ni l'intérêt d'apprendre des techniques d'analyse difficiles et complexes.

Les différents domaines d'applications de la simulation sont résumés dans le tableau suivant :

Région	Domaine d'application de la simulation
Systèmes informatiques	Les composantes «hardware», les logiciels, le réseau du «hardware», les bases de données et la gestion, le processus d'information, la fiabilité des «hardwares» et des logiciels, etc.
Domaines manufacturiers	Systèmes de manutention, les lignes d'assemblage, les installations de production automatisées, les installations de stockage, les systèmes de contrôle d'inventaire, l'étude de fiabilité et de maintenance, le plan d'aménagement, le design des machines, etc.
Les affaires	Analyse des stocks et des commodités, la politique des prix, les stratégies de marketing, les études d'acquisition, les "cash flow", les prévisions, les alternatives de transport, la planification de la main-d'œuvre, etc.
Gouvernement	Les armes militaires et leurs utilisations, les stratégies militaires, planification de la population, l'utilisation des terres, la distribution des soins médicaux, la protection contre les feux, services de polices, etc.
Ecologie et environnement	La pollution des eaux et leur purification, contrôle des déchets, la pollution de l'air, le contrôle des empoisonnements, les prédictions du temps, les explorations minérales et leur extraction, les systèmes d'énergie solaires, etc.
Sociale et comportement	Analyse de nourriture/population, les politiques d'éducation, structures organisationnelles, analyse du système social, les administrations universitaires.
Biosciences	Les analyses des performances du sport, le contrôle des maladies, les cycles de vie biologiques, les études biomédicales, etc.

Tableau 2 : Les différents domaines d'applications de la simulation (GPA662, 2007)

Le tableau suivant résume les objectifs communs qui ont motivé l'étude par simulation dans le passé.

Sujet	Objectifs de la simulation
Evaluation	Déterminer les performances du design de systèmes proposés quand ils sont évalués sous des critères bien spécifiques.
Comparaison	Comparer des designs de systèmes compétitifs pour une fonction spécifique, ou comparer différentes politiques d'opérations ou de procédures proposées.
Prédiction	Estimer les performances du système sous certaines conditions.
Analyse de sensibilité	Déterminer lesquels des facteurs sont les plus significatifs dans l'affectation des performances d'un système.
Optimisation	Déterminer quelle combinaison de variables donne la meilleure réponse.
Relations fonctionnelles	Etablir la nature des relations entre les variables et la réponse du système.

Tableau 3 : Objectifs de la simulation (GPA662, 2007)

Enfin, on peut souligner que la simulation en tant que méthode ne doit en aucun cas être vue comme un simple travail de programmation sur ordinateur, et ce, quelque soit la complexité du système à analyser. Elle doit obéir à une approche méthodologique qui est en grande partie indépendante du logiciel et du matériel utilisés.

3.3 Les étapes d'une étude de simulation

La conduite d'une étude de simulation comprend douze étapes schématisées par l'organigramme la Figure 5 :

1. **Formulation du problème** : Cette étape permet de préciser le contexte de l'étude et la formulation du problème. Si cette formulation est faite par les personnes qui ont rencontré le problème, l'analyste doit faire très attention de façon à comprendre clairement le problème. Si par contre, la formulation est faite par l'analyste lui même, il est très important aussi que les personnes concernés (clients de l'étude pour lesquelles le problème doit être résolu) comprennent et acceptent la formulation proposée. Durant cette étape, il est suggéré qu'un ensemble d'hypothèses soit proposées, discutées et acceptées par l'analyste et le client. Il faut souligner qu'à ce niveau, on n'est pas sûr d'avance que la formulation soit très juste et il est tout à fait possible que le problème aura besoin d'être reformulé au fur et à mesure de l'avancement du projet.

2. **Fixation des Objectifs** : Une fois le problème formulé, il faudra définir les objectifs visés par le projet de simulation. Ceci comprend :
 - Quelles hypothèses veut-on vérifier ? et dans quel contexte ?
 - Le personnel qui sera requis,
 - Le matériel et les logiciels informatiques,
 - Les divers scénarios qu'on veut expérimenter, les sorties attendues,
 - Les coûts de l'étude ainsi que les temps requis.

3. **Construction du modèle** : Cette étape consiste à construire un modèle conceptuel qui est une abstraction du système réel. Ce modèle peut être vu comme un ensemble de relations mathématiques et logiques concernant les composants et la structure du système. On doit en principe commencer par un modèle de base simple et l'affiner au fur et à mesure de façon à obtenir le modèle répondant aux objectifs visés. On doit souligner l'importance d'impliquer le

client dans cette étape. Ceci augmentera la qualité du modèle obtenu et accroîtra la confiance du client quant à l'utilisation du modèle. Enfin, on doit également noter que la construction du modèle est une tâche délicate et dont la maîtrise ne peut se consolider que par la pratique sur des problèmes réels.

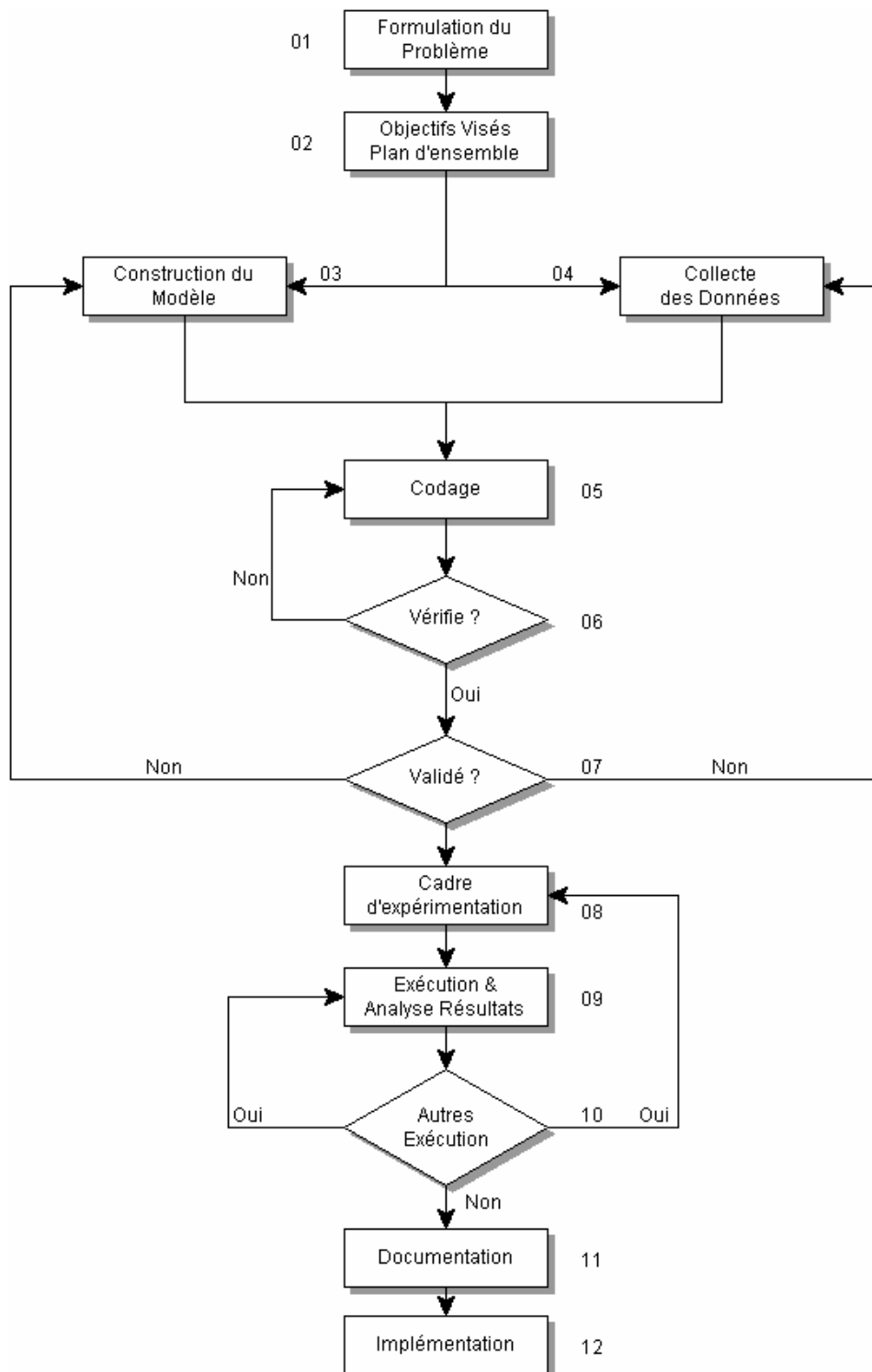


Figure 5 : Les étapes d'une étude simulation (Belattar, 2000)

4. **Collecte des données** : Cette étape consiste à spécifier et collecter les données dont on a besoin sur le système réel et les soumettre au client pour validation. Dans le meilleur des cas, si le client lui-même dispose déjà des données nécessaires (conservés dans le format adéquat), l'analyste n'aura qu'à les utiliser directement. Cependant, ce cas est vraiment très rare dans la pratique. En effet, très souvent le client pense disposer de toutes les données requises mais en réalité lorsque l'analyste prendra en compte celles-ci, elles s'avéreront très différentes de celles dont il a besoin. Il faut remarquer que sur l'organigramme, les étapes de construction du modèle et de collecte des données sont placées au même niveau. Ceci signifie que ces deux étapes peuvent progresser en parallèle.
5. **Codage** : Il s'agit de traduire le modèle conceptuel obtenu à l'étape 3 sous une forme acceptable par l'ordinateur (le programme est appelé aussi modèle opérationnel). Pour cela, il va falloir utiliser un langage de simulation parmi ceux disponibles. Certains langages de simulation offrent l'avantage de séparer entre le modèle et le cadre d'expérimentation qui contient toutes les données et les informations pour exécuter la simulation. Il devient alors possible de faire différentes expérimentations sur le même modèle en changeant uniquement le cadre d'expérimentation (données d'expérimentation). Il est important dans cette étape de construire un programme clair et facilement modifiable.
6. **Vérification** : L'étape de vérification est extrêmement importante dans tout projet de simulation. Elle concerne le modèle opérationnel (programme). Il s'agit de s'assurer que le modèle s'exécute sans erreurs. La vérification est primordiale même pour des modèles de taille réduite car ces derniers peuvent aussi comporter des erreurs bien qu'ils soient très petits comparés à des modèles de systèmes réels par essence complexes.
7. **Validation** : La validation consiste à s'assurer que le modèle conceptuel est une représentation fidèle du système réel. Il s'agit de savoir si le modèle proposé peut être substitué au système réel pour les objectifs de l'expérimentation. Dans le cas où le système existe, la façon idéale de valider le modèle conceptuel est de comparer ses sorties avec celles du système. Malheureusement, on n'a pas toujours cette possibilité, surtout dans les projets de conception de nouveaux systèmes. Dans ce cas, on utilise les techniques de validation utilisées en simulation. (Voire (Belattar, 2000))
8. **Conception d'un cadre d'expérimentation** : Dans cette étape, il s'agit de définir pour chaque scénario devant être simulé ou expérimenté un certain nombre de paramètres tels que : durée de

la simulation, nombre de simulation à faire (réplications), état initial du modèle, règles de gestion des files d'attente, etc.

9. **Exécution de la simulation et analyse des résultats :** Le modèle opérationnel ou programmé est le support principal pour réaliser une simulation sur ordinateur. Il sera analysé et interprété par le simulateur qui délivre en sortie des résultats purement statistiques (moyenne, variance, écart type, minimum, maximum, .. etc.). L'analyse de ces résultats aura pour objectifs d'estimer les mesures de performances du système conformément aux objectifs. .
10. **Exécutions supplémentaires :** à ce niveau, on dispose d'un ensemble de résultats provenant des différentes simulations ainsi que d'une analyse de ces résultats. Sur la base de cette analyse, on décidera si d'autres simulations doivent être réalisées, et si d'autres scénarios non prévus doivent être expérimentés afin de s'assurer que le modèle répond bien aux objectifs de l'étude.
11. **Documentation :** la documentation est nécessaire pour différentes raisons. Elle concerne aussi bien le modèle que les résultats de la simulation. Dans le cas où le modèle sera réutilisé par d'autres personnes, la documentation les aidera à mieux comprendre le fonctionnement du modèle et leur faciliter toutes modifications ou mises à jour de celui-ci. Aussi, si le projet a été réalisé pour un client spécifique, ce dernier sera plus satisfait et plus confiant lorsqu'il dispose d'une documentation claire sur le projet. En effet, il aura la possibilité de revoir toutes les alternatives prises en considération, les critères de comparaisons qui ont été utilisés et les recommandations faites par l'analyste. Ceci va l'aider énormément dans la prise de décision qui sera principalement basée sur les résultats fournis par la simulation et rapportés dans la documentation.
12. **Implémentation :** l'objectif de toute simulation est de proposer pour un problème donné plusieurs solutions possibles. Le choix de la meilleure solution devra être fait par l'analyste qui le justifiera dans la documentation et le proposera au client. La décision de retenir cette solution pour une éventuelle implémentation reste donc une responsabilité du client. C'est ainsi que la qualité et la richesse de la documentation peut avoir une grande influence sur cette étape. De plus, l'implication du client tout au long de la conduite du projet augmentera les chances d'une implémentation de la solution retenue.

A partir de cette description des étapes de l'étude de simulation ci-dessus, les observations importantes que nous pouvons faire sont : une étude de simulation est structurée en étapes qui pourraient être conduites dans un ordre prescrit. Chaque étape peut être conduite par une ou

plusieurs personnes et peut utiliser comme entrée, des résultats des étapes précédentes. Des documents sont produits à chaque étape (diagramme, rapport, code source, etc.), partagés par différentes personnes et peuvent mener à plusieurs versions. Il devient évident que tout collecticiel de simulation doit supporter en même temps : la communication, la coordination explicite et la coordination implicite à travers des documents partagés. Il faudra aussi tenir compte de la nature synchrone ou asynchrone de la communication entre les membres de l'étude. Ainsi, on peut dire que l'organisation de l'équipe va être un facteur important pour guider la conception d'un collecticiel de simulation.

3.4 Les outils de simulation

Définition

L'outil de simulation est l'ensemble des constituants d'un système (appelé généralement système de simulation) qui fournit les moyens de mettre en œuvre, d'animer et d'observer les modèles (Belattar, 2000).

Un système de simulation est connu dans la littérature du domaine sous diverses appellations telles que, logiciel de simulation, langage de simulation, simulateur, outil de simulation, etc.. . Afin de lever toute ambiguïté dans la suite de ce mémoire, nous utiliserons indifféremment l'une quelconque de ces appellations pour désigner un système de simulation.

Les caractéristiques de bases d'un outil de simulation

Les caractéristiques de bases que doit vérifier un bon outil de simulation et qui déterminent sa qualité, sont en générale (Hamadeh, 1993) :

- o Modularité : C'est la capacité de l'outil de simulation à prendre en compte une description modulaire du problème et ceci sous trois aspects : Décomposition (Ceci implique que l'outil de simulation prenne en compte naturellement la possibilité de décomposer un problème ou un modèle en sous problème), Composition (c'est la capacité du système à regrouper un ensemble de sous modèles en un modèle composé) et la réutilisabilité (la possibilité de la réutilisation de portions de programmes).

- Extensibilité : L'extensibilité mesure la capacité d'ouverture de l'outil de simulation à des situations nouvelles sans recours au concepteur du programme.
- Ergonomie : Au-delà des aspects d'interface utilisateur (programmation graphique ou textuelle), l'ergonomie est une caractéristique complexe à définir. La qualité recherchée ici est la facilité d'appropriation de l'outil informatique par l'utilisateur à tous les niveaux d'utilisation, y compris conceptuels.
- Cohérence de modélisation : C'est la possibilité que chaque objet dans un modèle ait plusieurs représentations selon la finesse de la modélisation et la granularité du temps. Exemple : lors de la phase de conception détaillée d'un système de production, un sous atelier composé de plusieurs postes de travail est représenté par un modèle qui met en évidence le flux de produit entre les postes de travail, ainsi que la communication et la synchronisation entre les différentes machines. Alors, on aimerait voir le modèle détaillé de l'atelier. Par contre, pendant la phase de conception générale, le modélisateur assemble des sous ateliers pour obtenir des ateliers, des ateliers pour obtenir un système de production. A ce niveau là, on n'a pas besoin de simuler le fonctionnement détaillé de chaque atelier. Les ateliers devraient avoir des fonctionnements approximatifs. Par souci que la taille du système explose, le logiciel de simulation devrait permettre à un objet d'avoir plusieurs représentations possibles dont chacune sera choisie selon la finesse de la modélisation. Dans ce cas, on peut interchanger cette représentation fine du sous atelier par une autre plus agrégée et on ne verra plus les composantes internes de ce sous atelier. Une pièce ou un chariot entrant dans le sous atelier sort avec des nouvelles valeurs correspondant aux temps d'attente, temps de passage, date de sortie... Ces valeurs dans la représentation agrégée peuvent être calculées approximativement.
- Autres exigences : Pour une utilisation aisée et un service de qualité, un simulateur comportera également : un éditeur graphique, autorisant la description et l'animation de synoptiques, une bibliothèque de programmes qui permettra une analyse et la représentation des résultats de la simulation.

3.5 Les logiciels de simulation

Nous pouvons distinguer deux étapes importantes lors de l'utilisation d'un logiciel de simulation. D'abord une phase de modélisation qui représentera l'architecture du système ainsi que son fonctionnement, d'autre part une phase d'interprétation et d'analyse des résultats (Figure 5).

On peut classer facilement, selon l'outil de base utilisé, les logiciels de simulation en trois familles : les logiciels dédiés écrits en langages traditionnels (Fortran, Pascal ou C), les logiciels de simulation utilisant un langage spécifique et les logiciels basés sur les réseaux de Pétri.

Niveaux d'utilisation

Il est commode de classer les utilisateurs d'un produit de simulation en trois catégories (Figure 6). Bien sûr il peut se faire que dans la réalité, ces utilisateurs ne soient pas complètement différenciés.

En général, au niveau 1 on utilise un langage général de type C. Au niveau 2 on utilise un langage simplifié (ou un langage de simulation) et au niveau 3, on a droit à un environnement type menu et fenêtre de dialogue.

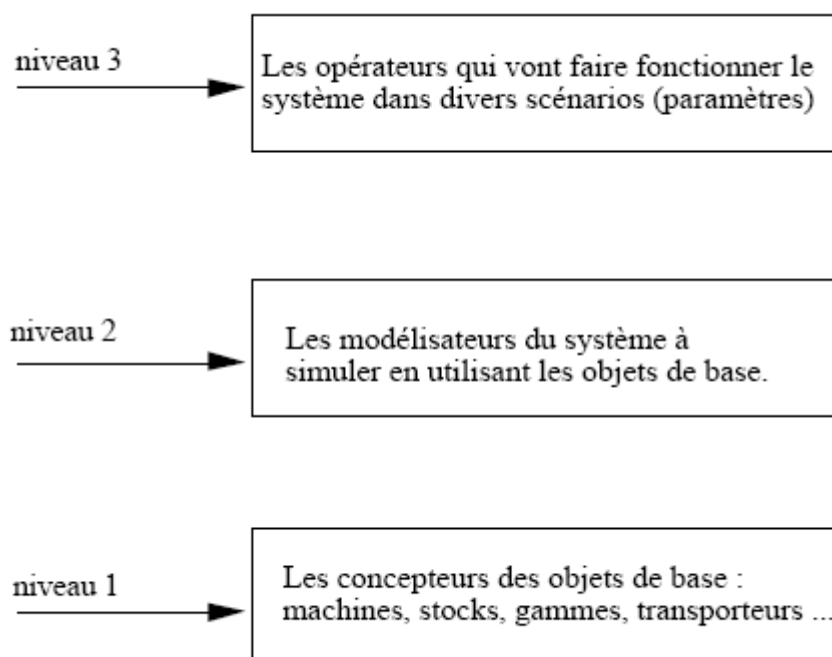


Figure 6 : Les différents niveaux d'utilisation (Hamadeh, 1993)

Logiciels dédiés écrits en langages généraux

Face à un type de problèmes particulier, l'utilisateur choisit un logiciel convivial et dédié à ce type de problèmes.

Un logiciel dédié permet à l'utilisateur de résoudre facilement un problème, parce que le programmeur du logiciel a déjà fait une partie de la modélisation, par exemple :

- il permet la description de la suite des opérations à effectuer par les postes de travail sur les pièces (les gammes).
- il permet d'associer des priorités aux machines par rapport aux moyens de transport....

L'utilisateur n'a qu'à manipuler des objets couramment utilisés dans son métier (machine, stockage, chariot, pièces,...).

Un logiciel dédié est souvent construit directement à partir d'un langage général (Fortran, Pascal ou C). Cependant, il existe des logiciels dédiés basés sur des langages spécifiques, ils présentent les mêmes qualités et les mêmes inconvénients au niveau de l'utilisateur final que ceux écrits en langages généraux.

L'inconvénient majeur de cette classe de logiciels concerne le nombre de type de problèmes qu'on peut résoudre. Face à un type de problèmes qui sort du cas standard, l'utilisateur se tourne vers un autre logiciel dédié ou fait recours au programmeur afin de lui apporter les modifications nécessaires.

On peut citer comme exemple de ce type de programme SIMFLEX (Simulation orientée ateliers flexibles), C.F.M. (simulation de chaînes flexible de montage) développés à l'INRIA par le groupe Systèmes de Production, OASYS et PARSIFAL (simulation d'ateliers de fabrication par lots) développé au Centre d'Études et de Recherches de Toulouse (CERT/DERA).

Logiciels utilisant un langage spécifique

Ces logiciels tentent de construire un outil convivial et extensible. Un outil est dit convivial s'il n'exige pas un effort particulier d'apprentissage et est dit extensible s'il permet l'ajout de nouvelles fonctions non prévues au départ.

On construit, avec un langage traditionnel, des fonctions pour l'utilisateur (par exemple: saisir une ressource) paramétrables. On définit un certain nombre de fonctions, plus ou moins abstraites qui constituent avec les règles de synthèse associées un langage spécifique pour la simulation des systèmes. La plupart de tels langages se caractérisent par une symbolique graphique d'une part, une grammaire (comme tout langage de programmation) d'autre part. Le passage du système graphique au langage est souvent automatisé (c'est le cas par exemple pour les langages SLAM et SIMAN).

Un tel langage spécifique constitue un produit très ouvert, pouvant traiter des problèmes aussi différents que l'assemblage ou l'étude des flux d'information dans un réseau, mais pas toujours très convivial car il exige de l'utilisateur un effort (apprendre le langage et l'utiliser en l'adaptant à son problème particulier).

Le constructeur peut décider d'aller plus près de l'utilisateur pour lui offrir un outil convivial. En utilisant un langage spécifique, il développe une couche logicielle dédiée à un certain secteur d'activités. Cette convivialité coûte souvent très cher, l'outil obtenu n'est plus extensible au niveau de l'utilisateur.

Les résultats sont le plus souvent présentés sous formes de tables et de graphiques (histogramme et camembert). Certains, grâce à un générateur d'icônes, offrent des possibilités d'animation (par exemple : SLAM et SIMAN).

La figure 7 montre le cheminement usuel pour simuler un système en utilisant un langage de simulation :

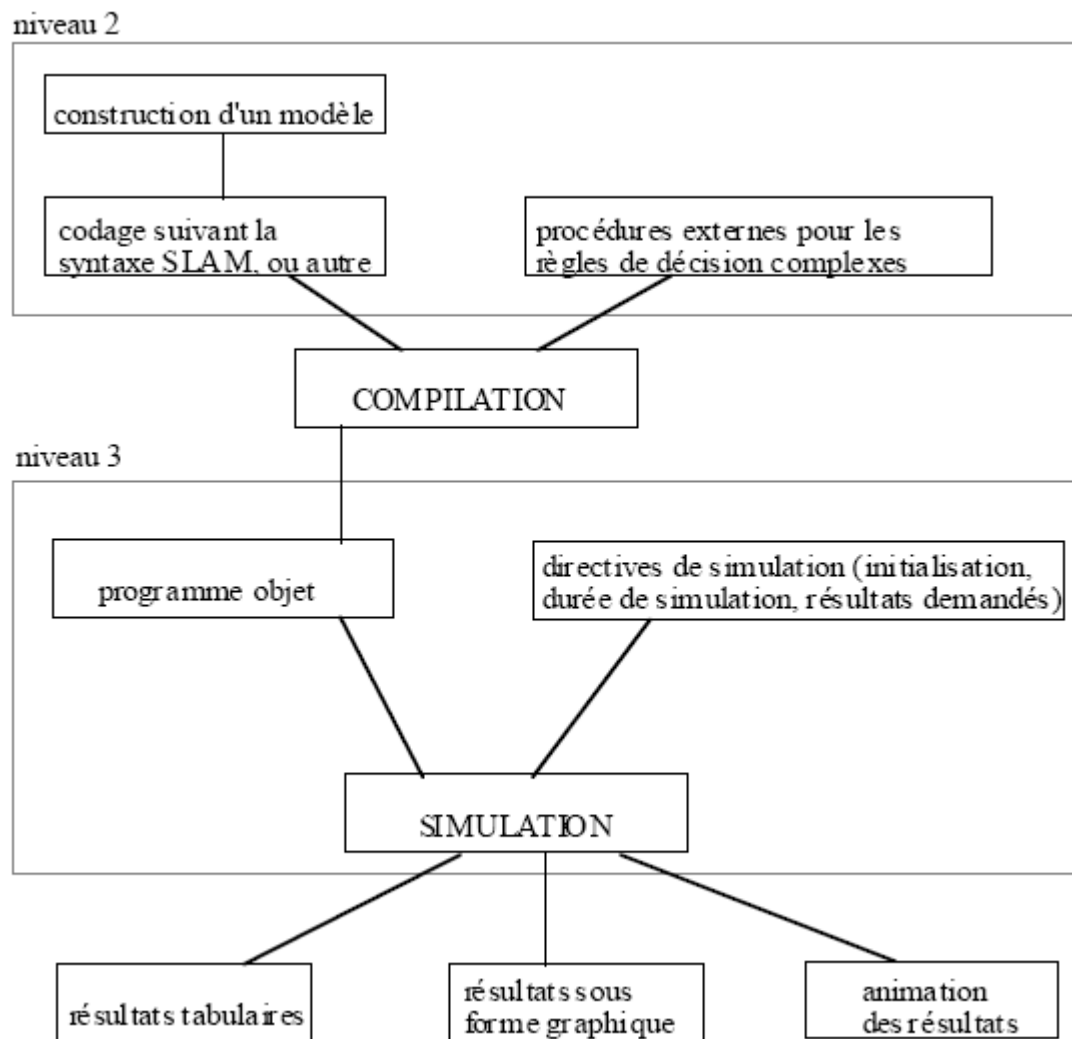


Figure 7 : le cheminement usuel de la simulation en utilisant un langage de simulation (Hamadeh, 1993)

La figure précédente (Figure 7) montre aussi les deux derniers niveaux d'utilisation : le niveau 2 qui consiste à l'utilisation du langage SLAM dans la construction du modèle et le niveau 3 qui consiste à ajuster certains paramètres et à lancer la simulation. Pour le même exemple, les utilisateurs du niveau 1 sont les constructeurs qui ont programmé le langage SLAM.

Logiciels basés sur les Réseaux de Petri

A la différence des simulateurs présentés ci-dessus, les simulateurs basés sur les Réseaux de Petri (RdP) permettent une analyse du modèle pour vérifier la cohérence par rapport au système physique. On peut citer comme exemple de ce type de simulateurs développés autour de modèles RdP : LORIC, SEDRIC et SICLOP.

L'inconvénient majeur des RdP concerne la représentation des données. En effet, un RdP décrit la structure de contrôle mais pas la structure de données. Il définit l'enchaînement des opérations effectuées sur les données mais pas ces opérations elles mêmes.

EXTEND : un progiciel de simulation moderne

Extend est une plate-forme de simulation idéale pour l'étude du comportement de systèmes complexes. Le logiciel s'applique en productique, logistique et organisation d'entreprises (tension des flux, dimensionnement d'investissement, en-cours, implantation, taille de lots, etc.), ainsi que pour les études de dynamique des systèmes.

Les systèmes étudiés sont représentés graphiquement grâce aux bibliothèques standard ou spécifiques à un domaine. Le langage de développement permet d'étendre les possibilités de l'environnement et de créer les objets spécifiques à un secteur d'activité (bibliothèque métier).

Extend fournit les composants pour modéliser les goulets d'étranglement, les aléas, l'utilisation des ressources, les règles d'exploitation, pour mesurer la productivité et la rentabilité du processus modélisé.

4 Simulation et systèmes de production

4.1 Complexité du système de production

L'environnement actuel caractérisé par une compétition féroce, par une augmentation des exigences des clients et par le développement de technologies avancées, a forcé les compagnies à repenser la manière avec laquelle elles gèrent leurs affaires. Aujourd'hui, les consommateurs demandent une meilleure qualité, des produits personnalisés et des services à bas prix et n'apprécient pas d'attendre. Puisque les attentes des consommateurs continuent à croître selon la qualité, la personnalisation des produits, le prix et les délais de livraison, les compagnies donnent plus d'attention à la manière dont les produits et les services sont fournis et non seulement quels produits et services sont fournis.

Dans ce contexte, le défi des industriels est apporté par les facteurs suivants (Figure 8) :

- Les systèmes ont des cycles de vie courts dû aux besoins en changements constants et la prolifération des nouvelles technologies.
- Les Systèmes devenus de plus en plus complexes résultant de la disposition de technologies plus sophistiquées et d'une plus grande intégration des processus.
- Les Systèmes nécessitant une plus haute performance à cause de l'augmentation de la compétition et des exigences des clients.

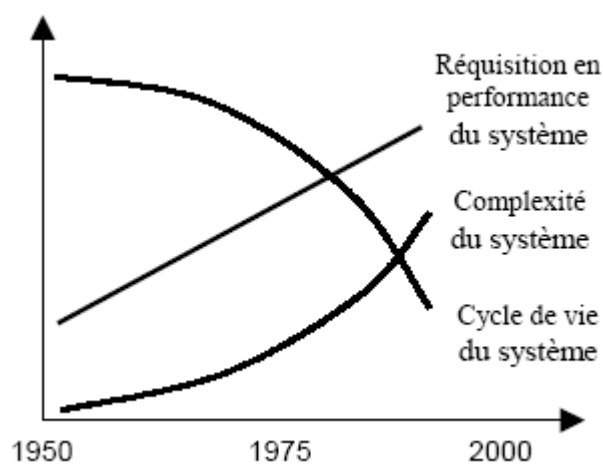


Figure 8 : Croissance du besoin en performance du système, augmentation de la complexité et réduction du cycle de vie (GPA662, 2007)

4.2 Limites des méthodes analytiques

Avec les défis en face des compagnies pour prendre des décisions plus difficiles et plus rapidement et qui ont un impact important sur la conception et l'opération des systèmes de productions, les ingénieurs et les gestionnaires sont à la recherche d'outils plus performants d'aide à la conception et à la planification opérationnelle des systèmes. Les méthodes traditionnelles tels que l'analyse du travail, les diagrammes de flux, l'analyse des processus, la programmation linéaire, etc. sont incapables de résoudre les problèmes complexes d'intégration d'aujourd'hui. Ces outils ont un champ d'application limité et sont incapables de fournir une mesure fiable de la performance espérée du système.

Les modèles analytiques ne représentent pas fidèlement la réalité puisqu'ils sont une agrégation (une simplification) de la réalité et souvent basés sur des hypothèses simplificatrices et non réalistes.

Généralement, les chercheurs qui développent les modèles analytiques ajustent souvent la réalité aux outils dont ils disposent. De plus, les modèles analytiques sont difficiles à valider au près des utilisateurs puisque ce sont des modèles synthétiques qui génèrent parfois des résultats non conformes à la réalité.

En l'absence d'outils performants de planification et d'évaluation, plusieurs compagnies optent pour les méthodes d'essais et d'erreurs qui se sont avérées assez coûteuses, consommant beaucoup de temps et assez perturbateurs pour fournir un bénéfice acceptable.

4.3 Apports de la simulation aux développements du domaine de production

Un outil qui a rapidement gagné une bonne réputation dans la conception et l'analyse des systèmes est la simulation par ordinateur. La simulation est un outil d'analyse puissant qui aide les ingénieurs et les planificateurs à prendre dans le temps des décisions intelligentes concernant la conception et l'opération d'un système. Par elle-même, la simulation ne résout pas les problèmes, mais elle les identifie clairement et évalue quantitativement les solutions alternatives. Comme un outil pour analyser "Quoi Si", la simulation offre des mesures quantitatives sur n'importe quel nombre de solutions proposées pour aider rapidement à restreindre la meilleure solution alternative.

En utilisant l'ordinateur pour modéliser un système avant sa conception ou bien pour tester des politiques opératoires avant qu'elles ne soient déjà mise en place, plusieurs pièges qui peuvent souvent être rencontrés lors du démarrage d'un nouveau système peuvent être évités. Les améliorations qui, auparavant, prennent des mois et mêmes des années à mettre au point peuvent être actuellement réalisées dans quelques jours et parfois en quelques heures en considérant la simulation par ordinateur.

L'habilité de la simulation à considérer un grand nombre et une large variété de variables dans un même modèle l'a rendu un outil indispensable dans la conception des systèmes complexes d'aujourd'hui (système manufacturier : variété des produits, des outils, des palettes, mécanismes de transferts, routes de transport, opérations, etc.)

Les bénéfices de la simulation plus semblables à celles réalisées par la simulation de vol où les pilotes gagnent de l'expérience sans courir le risque, le temps et le coût avec l'entraînement sur un équipement réel. Comme dans un simulateur de vol, la simulation des systèmes manufacturiers est effectuée pour améliorer la compréhension de l'utilisateur concernant comment le système opère de

telle sorte que des décisions intelligentes et habiles peuvent être prises, afin de réduire le temps et le coût associés avec l'expérimentation sur le système réel et de diminuer les risques d'erreur sur celui-ci.

En l'absence d'outils efficaces pour la planification et l'évaluation de systèmes, On a souvent recours à l'essai et erreur. Ceci est coûteux, prend du temps et perturbe le système. L'environnement change tellement rapidement qu'on ne peut apprendre des leçons. Donc on a grandement besoin d'une méthodologie prédictive basée sur la compréhension des causes et des effets. L'habileté de la simulation à considérer un grand nombre de variables dans un seul modèle fait d'elle un outil indispensable pour concevoir les systèmes de production complexes d'aujourd'hui.

4.4 Champs d'application de la simulation aux systèmes productions

La simulation est un outil versatile qui a été utilisé dans différents champs d'application incluant la conception des systèmes, la formation et l'éducation, la communication et les ventes et les relations publiques.

Conception des systèmes

Lors de la conception d'un nouveau système, des expériences peuvent être effectuées sur le modèle de simulation qui seraient autrement impossibles de réaliser sur le système réel puisqu'il n'a pas déjà été mis en œuvre. La simulation permet de réduire significativement le temps de déboguer (éliminer les erreurs) et de bien régler le système une fois installé.

Pour apporter des améliorations à système existant, la simulation permet d'effectuer l'expérimentation sur le modèle sans perturber le bon fonctionnement du système réel. Manipuler des objets et des ressources imaginaires offre une plus grande flexibilité d'apporter des changements et il est beaucoup moins coûteux que d'expérimenter le système actuel.

Lorsqu'elle est utilisée comme un outil de conception, la simulation traite des sujets tels que :

- la sélection des méthodes (les différentes activités devraient être effectuées sur une seule station ou bien divisée en plusieurs opérations?)
- la sélection de la technologie (quel est l'effet de l'utilisation de l'automatisation comparée à un procédé manuel?)
- optimisation (quel est le nombre optimum de ressources qui réalisent les objectifs au niveau performance?)

- analyse de la capacité (quel est la capacité de production du système?)
- décisions du système de contrôle (quelles tâches devraient être assignées et à quelles ressources?).

La simulation est un outil d'analyse et de vérification qui devrait être utilisé le plutôt possible afin de déceler les erreurs avant qu'elles ne soient coûteuses. Une règle importante de production appelée la règle des dix, mentionne que "le coût d'une erreur non décelée augmente d'un facteur de dix avec chaque opération jusqu'à qu'elle soit détectée" (GPA662, 2007). Adoptée à la planification des systèmes, cette règle dit que "le coût d'apporter une amélioration à un système donné augmente par un facteur de dix avec chaque stage additionnel".

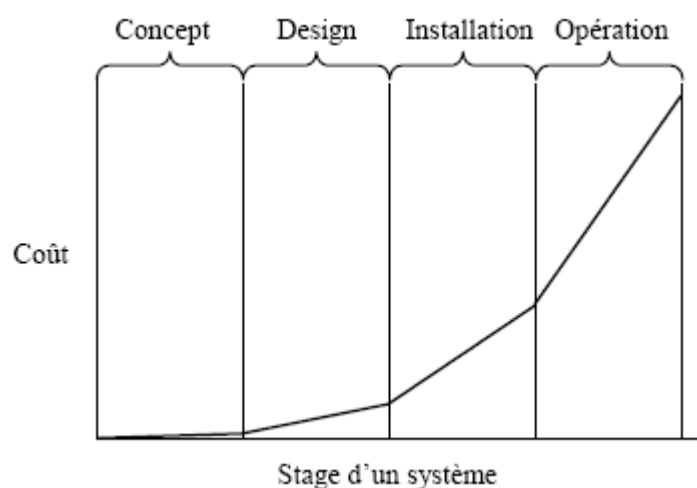


Figure 9 : Coût de changement effectué à chaque stage de développement d'un système (GPA662, 2007)

L'idée derrière la simulation consiste à effectuer le plus de changement possible durant le début des stages de conception et de design d'un système où le coût de changement demeure au minimum. La simulation assure aussi que les détails opérationnels du système sont considérés au début des stages de design de telle sorte que les questions qui se posent plus tard ont déjà été résolues (Figure 9). Il est à noter que ces détails ne sont pas souvent pris en compte lors de la conception des systèmes complexes.

Gestion des systèmes

Dans la gestion de l'opération d'un système, la simulation aide à déterminer le meilleur moyen pour contrôler le flux des clients et de matières. Elle aide aussi à trouver le moyen le plus efficace

pour l'ordonnancement et le déploiement des ressources. La simulation remplace les pratiques non fiables et de gaspillage de gestion basées sur les méthodes d'essais et erreurs. En simulant différentes alternatives de cédules de production, de politiques d'exploitation, de niveaux du personnel, de priorité des lots, de règles de décision, etc., le gestionnaire peut prédire avec plus de précision les sorties du système et peut ainsi prendre des décisions plus intelligentes et plus informées. En gestion des systèmes, la simulation assiste dans la prise des décisions suivantes :

- Ordonnancement de la production et des clients (quel est la meilleure séquence et le meilleur moment pour introduire les produits ou pour l'admission des clients dans le système?)
- Ordonnancement des ressources (quel ouvrier et quel équipement sont requis et à quel quart de travail?)
- Ordonnancement des travaux de maintenance (Quelle cédule de maintenance préventive qui perturbe le moins l'opération du système?)
- Priorité de traitement des travaux (Quel est l'ordre de priorité des tâches qui maximisent l'effort?)
- Gestion du flux (Quel est le meilleur moyen pour garder le flux de matières ou des clients dans le système le plus étendu et le plus déployé?)
- Gestion des délais et des inventaires (Quel est le moyen le plus efficace pour garder les temps d'attente des clients ou le niveau d'inventaire au minimum?)
- Gestion de la qualité (Quel est l'impact sur les opérations si les points d'inspection sont éliminés et le personnel doit assumer la responsabilité totale de la qualité de leur travail?)

La formation et l'éducation

La simulation peut aider les opérateurs, les représentants des services et les superviseurs à apprendre comment le système fonctionne, et qu'est-ce qui se passe si des décisions de gestion alternatives ou des politiques d'exploitation diverses sont mises en application.

La simulation est actuellement largement utilisée en éducation pour aider les étudiants à comprendre les interactions complexes qui existent dans différents systèmes de service ou manufacturiers.

Une approche de simulation qui est relativement nouvelle et prometteuse pour l'apprentissage et l'éducation est "la modélisation interactive" qui permet à l'étudiant d'intervenir lors de l'exécution du

programme de simulation afin de prendre différentes décisions. Les participants peuvent travailler individuellement ou en groupe pour tester leur habilité et traiter les sujets et les problèmes opérationnels associées avec différents systèmes.

Communication et ventes

Dans les systèmes de communication et des ventes, l'animation affichée durant la simulation offre un excellent aide visuel pour démontrer comment le système fonctionne. La simulation animée est assez convaincante dans la démonstration de l'efficacité d'un concept lors de la vente d'une solution particulière à un consommateur ou à un client.

La simulation permet de soulager et d'alléger les gestionnaires en leur offrant des preuves visuelles et quantitatives concernant le bon fonctionnement d'un système. Seulement un graphique unique est mieux qu'un millier de mots, et de même, une animation graphique est mieux qu'un millier de graphiques.

Relations publiques

La simulation permet d'élever l'image de la compagnie aux yeux du public large ainsi qu'aux yeux de ces clients. Elle transmet un message au monde extérieur indiquant que la compagnie est fière du fonctionnement de son système.

Dans le passé, les compagnies affiches des histogrammes et des graphiques de leurs ateliers situés stratégiquement à l'entrée de la firme. Dans le future, ces modèles seront remplacés par des animations en trois dimensions accompagnés avec le son et qui prennent les visiteurs dans un tour guidé des installations.

4.5 Exemples d'utilisation des logiciels de simulation dans le domaine de production

La simulation s'impose comme un outil d'aide à la décision sur tous les niveaux de conception d'un système de production. A tous ces niveaux, on retrouve le schéma traditionnel de la simulation, à savoir :

- réalisation d'un modèle correspondant à un élément précis du système étudié (modèle mathématique, symbolique et graphique, ou pictural),
- étude de ce modèle (étude des propriétés intrinsèques, étude analytique du comportement dynamique, simulation du modèle).

Dans la constitution des modèles, on distingue traditionnellement trois approches différentes :

- par événements : on étudie le système à travers ses changements d'état,
- par activités : on étudie les activités en cours à un instant donné,
- par processus : on identifie le système à un réseau de processus prédéterminés et connus.

Cette distinction semble un peu dépassée par l'apparition des outils associant les différentes approches. De plus, ces différentes approches font intervenir uniquement l'aspect fonctionnel du système, la notion de structures de données leur est totalement inconnue.

La modélisation et la simulation ont en fait trois types d'objectifs (quantitatifs, qualitatifs et stratégiques) associés aux différentes phases de mise en place des moyens de production.

On peut associer des outils aux différents niveaux de conception. L'utilisation des outils différents à chaque niveau de conception pose la question de communication entre les résultats de la simulation des différents niveaux.

Le niveau conception générale

Au niveau conception générale, le but est d'avoir des données quantitatives sur les flux de pièces, la charge des systèmes de transport, les emplacements de stocks à prévoir, le nombre de moyens de production à installer, la réaction du système face à des impondérables, l'évolutivité de ce système, etc... Les outils utilisables à ce niveau sont :

1. la modélisation mathématique (que ce soit avec la théorie des files d'attente, les chaînes de MARKOV, etc...) permettant l'analyse stochastique du processus. Un modèle mathématique peut rapidement amener des résultats intéressants en faisant abstraction des règles de décision et en faisant des hypothèses simplificatrices.
2. l'utilisation d'un langage de simulation à événements discrets (Q-GERT, SLAM, QNAP, SIMAN, etc...) Permettant la simulation du fonctionnement du système. Une simulation permet de valider ou améliorer les choix effectués et un certain nombre de décisions déjà prises.

En général, les modèles mathématiques et les langages de simulation ne sont pas concurrents, mais le plus souvent complémentaires.

Ces langages de simulation permettent la description de la structure du système par des graphes en utilisant des nœuds prédéfinis (nœud file d'attente, nœud de réservation de ressource, etc...) et paramétrables, les arcs entre ces nœuds ont aussi des fonctions déterminées (activités par exemple). Ces langages ne permettent pas la définition d'un nouveau type de nœuds. Une fois le modèle construit, ces langages permettent de faire circuler dans le réseau des entités typées, et de faire une analyse des résultats et des calculs statistiques.

Ces langages ne sont pas accessibles aux utilisateurs de tout niveau. On construit souvent une sur-couche logicielle permettant un accès facile aux utilisateurs non spécialisés dans la programmation.

Le niveau conception détaillée

Au niveau conception détaillée, le but de la simulation est de vérifier le bon fonctionnement des équipements mis en place. La simulation du fonctionnement du système se décompose en deux classes : une simulation d'ordre mécanique, et une simulation d'ordre plutôt logique.

- **Simulation mécanique** : La simulation d'ordre mécanique s'intéresse essentiellement à la capacité physique des équipements de travailler dans de bonnes conditions (étude de trajectoires, problèmes d'accessibilité, problèmes d'évacuation, etc...). Pour faire cela, on utilise donc des logiciels conçus à partir des logiciels de CAO 3D. Ces logiciels permettront de modéliser les équipements en question et de les animer avec possibilité de : déplacement, rotation, zoom, 4) ralentissement, accélération, etc...
- **Simulation de la logique de pilotage** : La simulation de la logique de pilotage s'intéresse aux protocoles de communication entre équipements, aux règles de synchronisation, aux procédures de reprise, etc... Le but est de vérifier la sécurité de leur fonctionnement. Les outils utilisables à ce niveau sont grafcet et Réseaux de Petri qui permettent une modélisation graphique des protocoles ou des procédures de reprise et d'étudier les propriétés de ces systèmes (absence de "dead-lock", etc..).

Le niveau fonctionnement

A ce niveau, les objectifs de la simulation sont d'ordre stratégique. Le but est alors d'examiner des orientations possibles :

1. des décisions de prise de commandes ou de marchés,
2. des décisions d'abandon de commandes ou de marchés,
3. des perturbations dans les approvisionnements,
4. des pannes ou immobilisations volontaires des moyens,
5. des modifications de gammes,
6. etc...

Les logiciels de GPAO comportent de plus en plus souvent des outils de simulation stratégiques. Ces outils sont de plus en plus des outils interactifs, conviviaux et permettent à l'utilisateur de forger ses propres décisions.

5 La conception coopérante d'un système de production

La conception des systèmes de productions complexes fait intervenir de nombreuses personnes avec des compétences et des rôles différents. En effet, dans le domaine de simulation tout projet de simulation quelle que soit son envergure fait intervenir de près ou de loin, de nombreuses personnes avec des compétences et des rôles différents (Statisticien, Informaticien, Industriel, Ingénieur de méthodes, Programmeur, Client de l'étude, etc.....). Jusqu'à une époque récente, les outils informatiques ne prenaient pas en compte cette dimension et laissaient au chef de projet le soin de gérer le processus de conception et sa dimension multi-participant hors système informatique.

Les recherches en travail coopératif assisté par ordinateur ont pour but de proposer des collecticiels permettant aux hommes de réaliser une tâche "en commun" à partir de leurs postes de travail respectifs ou depuis des installations spécialisées (ex: salles de vidéoconférence). Sur un plan informatique et social, il s'agit maintenant de gérer non seulement l'interface homme-machine mais également l'interface homme-homme médiatisée par la machine.

Dans un collecticiel, la relation entre les différents participants peut être envisagée sous différents angles. On peut ainsi vouloir réunir des personnes distantes géographiquement (bureau à côté, dans une autre ville, pays ou continent) ou ne travaillant pas en même temps (rythmes

différents, emplois du temps incompatibles, décalages horaires, etc.). Dans tous les cas, on constate qu'il s'agit de gérer, via le système informatique, la participation de plusieurs personnes qui peuvent n'être présentes que virtuellement (dans l'espace et/ou dans le temps). Le premier objectif des collecticiels est donc de proposer un support d'abolition des dimensions espace et temps.

Les collecticiels doivent aussi prendre en compte les aspects organisationnels du travail. Ainsi, les intervenants constituent des groupes de travail qui doivent s'organiser et se situer dans le temps et dans l'espace. Ils sont amenés à définir des rôles, des sous-groupes et des phases de travail.

Le succès d'un travail coopératif peut se mesurer à la façon dont le collecticiel est capable de créer, et de soutenir une bonne dynamique de groupe. Celui-ci doit donc contribuer à faire disparaître la virtualité de présence des participants. Le travail doit pouvoir se dérouler au moins aussi naturellement qu'en présentiel et sans le support informatique. Il doit même bénéficier d'une organisation du travail plus efficace s'appuyant sur les nouvelles possibilités offertes par l'informatique. Celle-ci constitue ainsi un nouveau support de production, de conversation, de communication et de coordination entre les hommes. Le dispositif technologique mis en place ne doit pas perturber le travail ni la dynamique de groupe créée. Lors de la conception de systèmes coopératifs, il faut donc avoir conscience que la dimension usage, qui permet de valider l'environnement proposé, est au moins aussi importante que la dimension réalisation.

Cette problématique générale du travail coopératif est parfaitement applicable à l'activité de conception d'un système de production. En effet, il est possible de bâtir différents scénarios de collaboration entre concepteurs et de modéliser la conception coopérante. Celle-ci se justifie par le fait que la conception d'un système de production complexe nécessite des compétences pluridisciplinaires très poussées, si bien que personne ne peut s'acquitter seul de cette tâche. De ce fait, les problèmes sont découpés en études de sous-ensembles possédant entre eux des relations et des contraintes. Ces études sont attribuées à plusieurs personnes dont la tâche commune est de réaliser in fine un système global constitué par l'ensemble des sous-systèmes. C'est en ce sens que l'on peut parler de travail coopératif. Les sous-ensembles correspondent à des spécialités éventuellement différentes. Toutefois, les interactions définies entre eux sont suffisamment fortes pour que les choix effectués par un intervenant aient des conséquences pour les autres.

Théoriquement un travail en groupe peut s'organiser selon les deux modes extrêmes de travail que sont la sous-traitance et la co-traitance. Le mode de travail effectif est souvent une combinaison de ces deux modes (Tarpin-Bernard, 1997b).

Dans les deux cas, le chef de projet décompose le travail à réaliser en parties et le décrit sous forme d'objectifs et de contraintes. Il fournit le sous-ensemble de départ et le contexte à chaque participant ou équipe. Chaque participant travaille sur son sous-ensemble en respectant les objectifs et les contraintes imposés puis fournit au chef de projet le résultat du travail. Le chef de projet réceptionne les sous-ensembles réalisés, les contrôle et effectue la fusion.

Dans le mode de sous-traitance, toutes les contraintes doivent être parfaitement définies afin que chaque participant puisse travailler de façon autonome. La coopération se limite aux interactions ponctuelles et aux phases de validation et de transmission des informations.

Dans le mode de co-traitance, les interactions entre participants ont lieu durant tout le processus. Celles-ci permettent de détecter et de résoudre les problèmes au fur et à mesure. D'une façon théorique, lors d'un travail à plusieurs personnes, les phases d'analyse, de conception et de validation peuvent être menées soit de façon séquentielle (pure ou itérative), soit de façon concurrente comme la montre la figure 10. Bien sûr, la co-traitance ne peut pas s'effectuer en mode séquentiel pur ; il n'y aurait pas d'interactions possibles entre les participants.

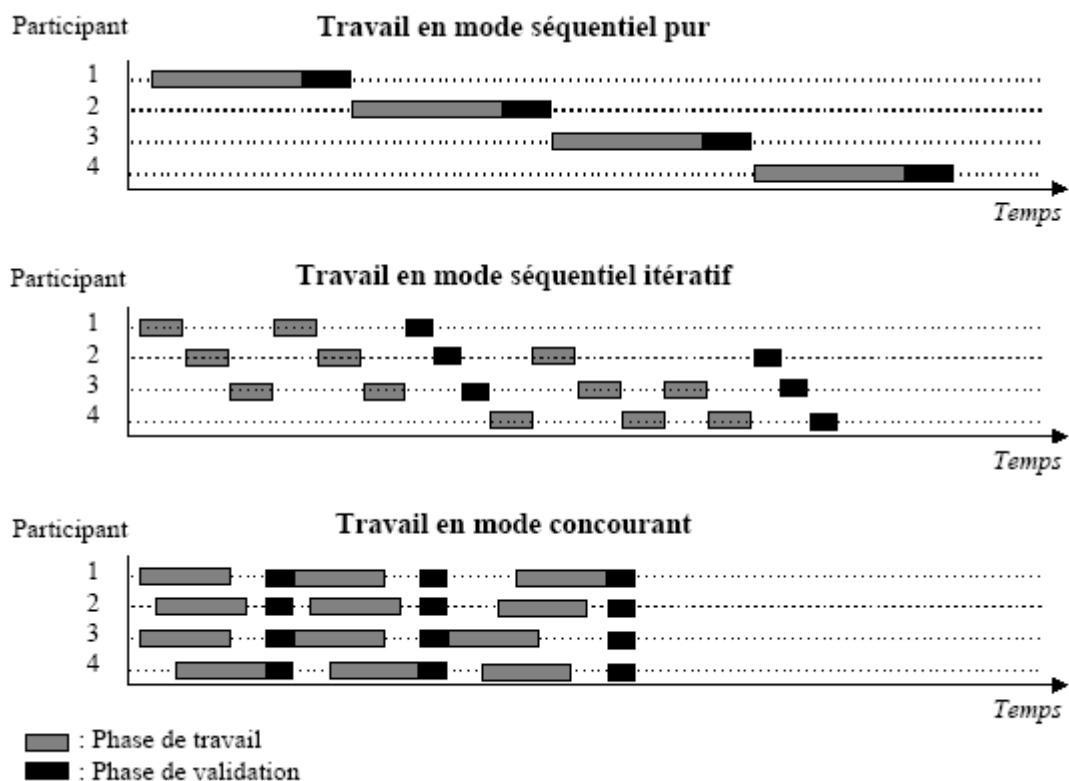


Figure 10 : Différents modes d'organisation du travail coopératif. (Tarpin-Bernard, 1997b)

Dans le cadre d'une conception d'un système de production, le travail à mener constitue un projet placé sous la responsabilité d'un chef de projet. Dans ce cadre, le chef de projet a le rôle d'animateur. A ce titre, il se porte garant de la cohérence et du bon avancement du projet. Après avoir décomposé l'ensemble à réaliser en un certain nombre de sous-ensembles, il identifie les droits et devoirs qui incombent aux différents participants auxquels il confie l'étude détaillée et la réalisation de chaque sous-système. Il donne ensuite les contraintes, et suit les travaux tout en arbitrant les éventuels conflits entre participants. Pour finir, il valide les sous-ensembles en s'assurant qu'ils sont aptes à réaliser exactement les tâches définies par leurs spécifications. Si tel est le cas, il place les travaux dans un environnement de référence protégé. De ce fait, il assure l'unicité des versions du travail du groupe. Chaque participant peut néanmoins archiver, dans un environnement privé, une version de ses travaux différente de celle retenue par l'animateur.

Pour les autres participants de projet chacun d'entre eux possède un certain degré de liberté dans la conception et la réalisation du travail qui lui est confié, mais il doit connaître le contexte de son intervention, c'est-à-dire les liaisons d'interdépendance avec les autres sous-ensembles et avoir une vue générale du système pour situer les tâches des autres co-concepteurs. L'attribution des rôles reflète l'organisation du groupe dans un contexte de travail particulier.

Hormis le chef de projet qui peut être assimilé à un donneur d'ordres, on peut citer un certain nombre de rôles génériques tels que : exécutant, évaluateur, validateur, consolidateur et conseiller (expert). Dans certaines phases particulières telles que les réunions, on voit aussi émerger des rôles nouveaux comme modérateur, organisateur, etc.

6 Entreprise-projet de simulation

Le dictionnaire le Robert de la langue française (Le petit Robert 1992) définit le projet comme étant « l'image d'une situation, d'un état que l'on pense atteindre ». Les auteurs du dictionnaire précisent cette définition en indiquant qu'un projet est « tout ce par quoi l'homme tend à modifier le monde ou lui-même, dans un sens donné ». Les corollaires mentionnés montrent qu'il existe plusieurs acceptions du terme de projet : Dessin, intention, plan, résolution, vue (projet de mariage, projet personnel) ; mais aussi esquisse, canevas, ébauche (rédiger un projet de thèse, projet de loi) ; ou encore plan, programme (projet économique, administratif).

Dans le contexte industriel, un projet est caractérisé, selon l'AFITEP (Association Française des Ingénieurs Techniciens d'Estimation de Planification et de Projet), par «une action spécifique,

nouvelle, qui structure méthodiquement et progressivement une réalité à venir pour laquelle il n'y a pas d'équivalent exact» (Hanser, 2003). L'activité de projet possède un objet unique et ne peut être répétée à l'identique car chaque projet possède un contexte et une histoire propre. Un projet n'est donc jamais un constat ou la répétition d'un modèle préalablement établi, même si dans le cadre industriel, il arrive que l'on porte un regard très procédural sur l'activité de projet : « le projet est un ensemble d'actions à réaliser pour satisfaire un objectif défini, dans le cadre d'une mission précise, et pour la réalisation desquelles on a identifié non seulement un début mais aussi une fin ». Cette esquisse de définition montre que le concept de projet s'applique à différents domaines dont les particularités influent sur le caractère opératoire du projet. Les caractéristiques que nous venons de dégager montrent que le projet est un ensemble d'activités de conception situées et contextualisées. La nature du concept de projet tend donc à l'opposer à celui d'une activité stabilisée telle que la production, la vente, l'administration, etc.

Nous venons de voir que le terme de projet s'applique à différents types d'activités. Cependant, l'appréhension d'un projet se fait par l'identification de trois degrés (Hanser, 2003) : un degré empirique (concerne les situations de la vie quotidienne), un degré théorique (discours scientifiques où le projet a un statut théorique et conceptuel) et un degré opératoire (lié aux nécessités d'une action à conduire) ; ainsi on peut juger qu'un projet de simulation est de dimension conceptuelle, détaché de toute volonté de concrétisation.

Le concept d'entreprise-projet, aussi appelé entreprise fugace ou éphémère, opposé au concept traditionnel d'entreprise. À l'inverse de celle-ci, l'entreprise-projet n'est pas une entité juridique pérenne. Elle désigne une équipe dont les membres sont issus de plusieurs organisations et réunis à l'occasion d'un projet éphémère ; la durée de vie de l'équipe est la durée de vie du projet. Certes, ce concept n'est nouveau que dans le langage mais il marque le début d'une gestion nouvelle du travail et de l'équipe, comme si ses membres appartenaient à la même entreprise. Ce qui compte alors, c'est la place de chacun dans cette entité virtuelle plutôt que dans leurs entreprises respectives. L'entreprise-projet s'appuie sur la mise en place de nouvelles formes de division du travail, de nouvelles formes de contrat (contrats de réseaux), de la promotion de nouvelles valeurs et de nouvelles cultures d'entreprises. Il nous semble que ce concept est en accord avec les caractéristiques d'un projet de simulation.

Depuis quelques années, une nouvelle génération d'environnements ou d'outils logiciels, faisant usage du réseau Internet, est apparue, censée faciliter en particulier le travail coopératif dans le cadre de l'entreprise-projet ; ce sont les collecticiels (ou logiciels de groupe) dont l'exemple le plus connu est le gestionnaire de courrier électronique. Ces collecticiels de projet sont en accord avec les

caractéristiques de l'entreprise-projet : un réseau virtuel de personnes qui peut être simplement constitué, modifié puis défait ; de la même façon, une personne peut appartenir à plusieurs réseaux (c'est-à-dire participer à plusieurs projets) simultanément et ce, de façon très facile.

Dans la plupart des cas, ils sont d'un usage facile, et s'adressent à un public très large, non forcément spécialiste de l'informatique. D'autre part, fondés sur Internet, ces collecticiels sont susceptibles de toucher toutes les personnes qui disposent d'un moyen d'accès à ce réseau mondial. Certains collecticiels se destinent plus spécifiquement au projet coopératif ; ils permettent de partager des ressources communes comme des fichiers ou des messages.

En revanche, les modes de coopération sous-tendus par la plupart de ces collecticiels de projet, nous semblent davantage répondre à schémas hiérarchiques et autoritaires traditionnellement employés dans l'entreprise classique. Ces collecticiels génèrent une contradiction interne avec d'un côté, une technologie permettant une souplesse et une légèreté caractéristique de l'entreprise-projet, et de l'autre, des modes de fonctionnement trop souvent lourds et complexes.

6.1 Les acteurs d'un projet de simulation

Maîtrise d'ouvrage et maîtrises d'œuvre

- **Maître d'œuvre, maîtrise d'œuvre** (KLC, 2007) : Un maître d'œuvre est un professionnel responsable de la production d'un objet, ou de la fourniture d'un service. Le maître d'œuvre peut être externe ou interne à l'entreprise. La maîtrise d'œuvre est l'ensemble des fonctions exercées par le maître d'œuvre : on exerce la maîtrise d'œuvre d'un projet, d'une prestation. Par extension, on désignera également par « maîtrise d'œuvre » l'ensemble des personnes qui exercent la maîtrise d'œuvre.
- **Maître d'ouvrage, maîtrise d'ouvrage** (KLC, 2007) : Le maître d'ouvrage est le propriétaire de l'objet produit, ou le bénéficiaire des services fournis. Le rôle du maître d'ouvrage par rapport au maître d'œuvre est de choisir le maître d'œuvre, d'exprimer ses besoins, de valider les spécifications du produit à réaliser ou de la prestation à fournir, éventuellement de participer aux travaux, de suivre les travaux, de recetter la livraison, de payer le prix convenu. La maîtrise d'ouvrage est l'ensemble des fonctions exercées par le maître d'ouvrage : on exerce la maîtrise d'ouvrage d'un projet. Par extension, on désignera également par « maîtrise d'ouvrage » l'ensemble des personnes de l'entreprise qui exercent la maîtrise d'ouvrage.

- **Ouvrage et œuvre** (KLC, 2007) : Au sens retenu dans les concepts de maîtrise d'ouvrage et de maîtrise d'œuvre, l'ouvrage est un objet produit par le travail d'un professionnel, tandis que l'œuvre est le travail fourni par le professionnel pour réaliser l'objet. Dans le cas des projets informatiques, ces définitions s'appliquent parfaitement à l'ingénierie informatique, où le résultat du travail du prestataire est le système informatique, qui est un objet visible, tangible. Elles peuvent s'appliquer par analogie aux autres travaux (formation, mise en fonctionnement, étude ...).

Relations entre maîtrise d'ouvrage et maîtrise d'œuvre

On entend régulièrement des réflexions sur le caractère désuet, non-opérationnel de la distinction entre maîtrise d'ouvrage et maîtrise d'œuvre. Il est clair que, pour les projets informatiques, la relation entre maîtrise d'ouvrage et maîtrise d'œuvre n'est pas aussi simple que par exemple dans le Bâtiment : la spécification et la recette des systèmes sont plus lourdes et plus complexes, les techniques sont moins stabilisées, et dans beaucoup de phases du projet la maîtrise d'ouvrage est amenée à collaborer avec la maîtrise d'œuvre.

Ce qui est clair, c'est que dans les relations entre entreprises, il y a une entreprise cliente, qui est propriétaire du futur système d'information et qui le paie, et une ou plusieurs entreprises prestataires. Les contrats de travaux signés entre client et prestataires font apparaître des relations de type maîtrise d'ouvrage/maîtrise d'œuvre : expression des besoins, spécification du système à réaliser ou des services à fournir, recette ou contrôle des livraisons. Ces relations sont à adapter à la nature des systèmes informatiques, et des travaux : en particulier, quand l'entreprise cliente participe activement aux travaux de réalisation, comme pour le paramétrage d'un progiciel, la responsabilité du prestataire n'est pas de même nature que quand la réalisation est faite complètement par le prestataire.

Dans l'organisation interne de l'entreprise, les directions utilisatrices et la direction des systèmes d'information ont également des relations de type maîtrise d'ouvrage/maîtrise d'œuvre, avec le même type de réserves.

Les acteurs

Le collectif d'acteurs, dans sa taille et dans sa configuration, varie d'un projet à un autre. Cette variation n'est pas seulement liée à la taille du projet et à son coût, bien qu'elle dépende largement de ces deux facteurs, mais également à sa complexité fonctionnelle et technique; ainsi, la simulation d'un avion fait intervenir des dizaines de spécialistes et des personnes compétentes. À l'opposé, la simulation d'un moteur électrique ne requiert dans la majorité des cas que peu d'acteurs : un informaticien et un électrotechnicien.

On repère néanmoins pour tout projet trois familles d'acteurs déterminées par trois grandes fonctions comme le montre la figure suivante :

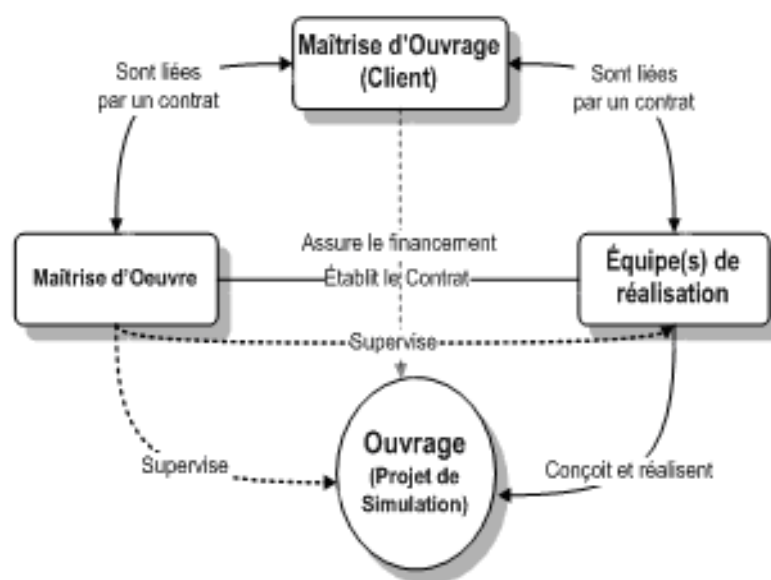


Figure 11 : Les familles acteurs d'un projet de simulation

- o La maîtrise d'ouvrage : qui est la personne physique ou morale pour le compte de qui est réalisé le projet ; il est généralement le futur bénéficiaire. En l'absence de compétences requises, le maître d'ouvrage s'adjoindra les services d'une ou plusieurs personnes qui seront alors reconnues comme maître d'ouvrage adjoints ou délégués. La maîtrise d'ouvrage, qui est représentée généralement par un ou plusieurs personnes de métier, se charge d'élaborer en amont un cahier des charges. Il est généralement élaboré en lien avec un expert en simulation qui apporte ses connaissances et son expérience des projets de simulation, en vue de juger de la faisabilité des demandes à la fois en termes technique et financier. Il peut également épauler le maître d'ouvrage dans ce processus d'analyse et l'accompagner par la suite lors de l'implémentation de la solution.

- La maîtrise d'œuvre : est la personne physique ou morale qui réalise le projet et qui assure la responsabilité globale de la qualité d'étude, du délai et des coûts. Dans le cas où la maîtrise d'œuvre est prise en charge par le maître d'œuvre elle peut être sous-traitée (entièrement ou en partie) à un prestataire de simulation. Cela intervient par exemple quand l'entreprise préfère éviter d'internaliser une activité ne constituant pas son cœur de métier.
- L'équipe de réalisation est celle que le maître d'ouvrage et/ou le maître d'œuvre a chargé par contrat de la responsabilité de l'ensemble de la réalisation du projet. Les principaux participants dans l'équipe chargée de la réalisation du projet sont : les informaticiens (Ingénieur spécialisé en simulation, Analyste, Programmeur, Agents de saisie... etc.), le personnel de métiers (Ingénieur de méthodes, Techniciens, Utilisateurs), les mathématiciens (Statistiques et Probabilités) et les experts consultants (experts en simulation, experts en métiers) comme le montre figure 12 ci-dessous.

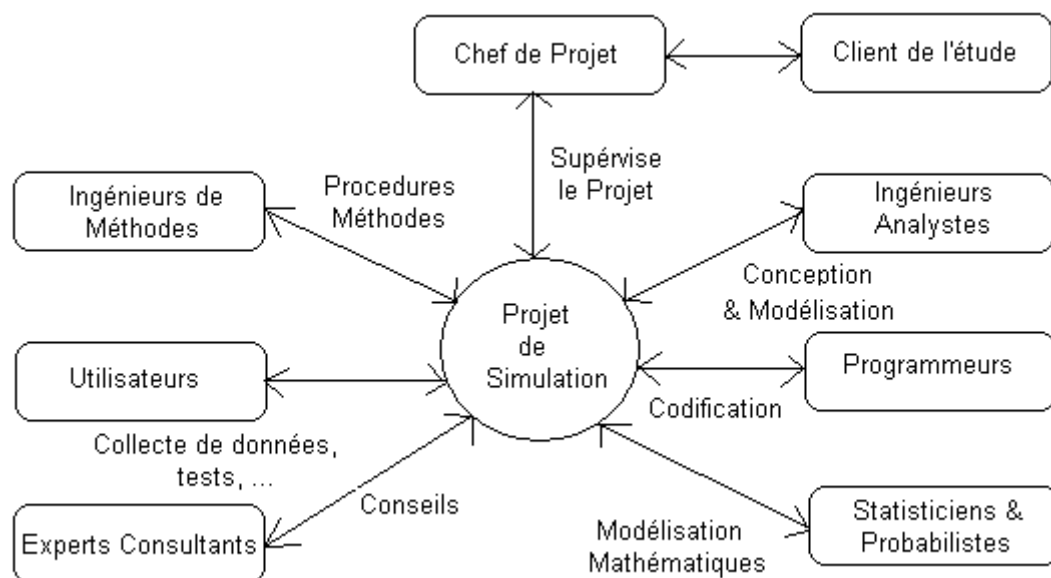


Figure 12 : Liste de participants d'un projet de simulation

6.2 Les moyens de coordination

La coordination est l'activité consistant à gérer les interdépendances entre activités. Elle est d'autant plus nécessaire que les activités partagent entre elles des ressources (espace, temps, personnes, outils, information, etc.).

Comme tout projet d'entreprise, la coordination entre les intervenants d'un projet de simulation peut se faire par deux moyens : les réunions, ou plus généralement les rencontres directes entre intervenants, et la planification par la mise en œuvre de techniques graphiques que sont les diagrammes de Gantt et les réseaux Pert.

Les réunions

Une réunion rassemble sur une durée relativement courte les intervenants essentiels aux activités en cours pour valider des choix et des décisions, puis décider de la suite de l'exécution de l'opération. L'organisation des réunions obéit généralement à une procédure formalisée ; les participants sont convoqués, un ordre du jour est établi, un rapport de réunion est envoyé à tout les participants, etc.

L'intérêt des réunions pour les acteurs vient justement du contact direct pour prendre rapidement des décisions. Leur importance sociale est telle qu'il serait difficile d'imaginer leur suppression, même si l'on sait que les technologies de communication permettent dans une certaine mesure de s'affranchir de l'inconvénient majeur des réunions, l'obligation de se déplacer. On repère au moins quatre fonctions assumées par les réunions :

1. une évaluation du travail effectué jusqu'alors, et par exemple depuis la dernière réunion,
2. une prise en compte d'éléments nouveaux susceptibles d'orienter la marche de l'opération,
3. une actualisation des objectifs des moyens et des méthodes, ainsi que la proposition d'une nouvelle date de réunion,
4. l'écriture et la distribution d'un rapport de réunion aux intervenants concernés.

Cependant, les réunions ne font pas l'unanimité parmi les professionnels. Les arguments avancés tendent à montrer que dans certaines circonstances la multiplication des réunions est un facteur de non-qualité. Deux arguments sont évoqués :

- o Le rythme élevé des réunions gêne le travail. Lorsque pour un seul projet, l'équipe de maîtrise d'œuvre se rencontre deux ou trois fois par semaine, sachant que les délais sont courts, nous comprenons la pertinence de l'argument.

- Les réunions convoquant un nombre élevé d'acteurs sont souvent moins prolifiques que les réunions « restreintes » parce que « tout ne peut pas se dire devant tout le monde ». Les précautions de langage augmentent avec le nombre d'interlocuteurs.

Techniques de planification (coordination programmée)

Dans un sens de contrôle de la coordination, la planification des tâches utilise deux techniques graphiques courantes que sont le diagramme de Gantt et le réseau Pert. La finalité de ce deux techniques est la même : coordonner les tâches afin de maîtriser les délais et les coûts.

Le réseau PERT montre davantage que le diagramme de GANTT la relation de dépendances entre les activités ; c'est en particulier cette caractéristique qui lui donne cette capacité à déterminer le chemin critique qui conditionne la durée maximale du projet. Le diagramme de GANTT permet quant à lui une bonne appréhension des durées relatives des activités et des recouvrements entre activités. Dans les deux cas, ces méthodes permettent de :

1. définir les activités constituant le projet,
2. organiser les activités dans le temps (ordonnancement),
3. évaluer les dépendances entre activités,
4. évaluer l'effort nécessaire pour chaque activité,
5. affecter les ressources (personnel, matériel) aux activités.

Les échanges d'informations

Un projet de simulation est le lieu d'un échange d'informations diverses. Les documents sont un type d'objet informationnel particulier, mais l'information produite, utilisée, transformée durant un projet ne se limite évidemment pas aux seuls documents. Une conversation téléphonique, le caractère urgent d'un envoi, ou l'expérience passée sur d'autres projets, sont des éléments d'information qui ont leur importance et que doivent prendre en compte les acteurs. Dans le cadre de notre recherche, nous employons le plus souvent la notion d'information dans un sens fini, celui de données.

Lorsque l'information prend une forme définie et traçable, on parle à la suite d'objets intermédiaires. Les documents en font bien évidemment partie. Les objets intermédiaires constituent une ponctuation dans le temps ; ils résultent d'une matérialisation momentanée des

interactions. Mais plus que cela, ils transforment l'action collective : par exemple, le document véhicule des demandes, des approbations ou des refus, infléchit des choix, repose la relation entre son émetteur et son destinataire, etc.

Le nombre et la nature des documents échangés au cours d'un projet est variables et dépend de l'envergure du projet. Leur répartition relative dans les différentes phases est aussi variable. Il est possible d'ordonner les documents en fonction de la diffusion dont il font l'objet :

- les inter-documents ; par extrapolation du terme d'intertexte, nous désignons par le terme d'inter-documents tous les documents qui véhiculent le savoir et l'expérience collective du domaine (livres, manuels, revues, catalogues, bases de données, etc.)
- les extra-documents ; ce sont les documents véhiculés entre participants lors déroulement du projet. Ils servent à échanger de l'information et à permettre le travail coopératif.
- les intra-documents ; ce sont les documents propres à un participant et nécessaires à l'accomplissement d'une tâche. Ces documents de travail n'ont pas a priori vocation à être échangés. Ce peut être les notes sur un cahier, etc.

Le mode d'échange de documents est la conséquence du support utilisé. Les documents papier par exemple, sont échangés et ne sont que rarement partagés. L'échange consiste à transmettre un document à l'occasion d'une rencontre entre participants ou par courrier (électronique ou postal) ; dans le cas des documents en papier, l'expéditeur prend préalablement soin d'en faire au moins une copie ; il transmet alors soit l'original soit la copie et conserve l'autre document.

Il existe toutefois des situations où le partage de documents entre les membres d'une équipe est nécessaire, par exemple dans le mode travail concourant. Le partage des documents papier est nettement moins répandu parce qu'il est plus difficile à mettre en œuvre. A l'opposé, le partage des documents électroniques est une possibilité nouvelle offerte par les nouvelles technologies. L'échange conserve cependant un atout auquel ne peut que difficilement prétendre le partage. Il s'agit du contrôle de la diffusion du document ; l'expéditeur sait à qui et à quel moment il a communiqué un document. Ce mode comporte deux avantages :

- Le destinataire ne peut ignorer le document directement transmis ; en cas de litige, l'expéditeur n'aura pas à se reprocher l'absence de prise en compte du document par ses partenaires.
- L'échange permet de sélectionner les destinataires pour chaque étape de l'évolution du document et une gestion différenciée des relations d'échanges en fonction de la plus ou moins grande proximité conceptuelle des membres de l'équipe.

6.3 La démarche

Quel que soit le domaine d'étude ciblé, la mise en œuvre d'un projet de simulation se décline en plusieurs étapes qu'on doit suivre avec le plus grand soin. Elle débute par des actions touchant à la formalisation des objectifs et du cahier des charges. Vient ensuite la planification (délais, finance, etc.), ainsi que l'acquisition du matériel et logiciel informatiques, le recrutement de personnes compétentes et l'attribution des rôles et objectifs dévolus à chacune. Lors des déploiements en tant que tels, la démarche se poursuit par la coordination et le suivi des équipes et des travaux qui peut d'ailleurs s'effectuer par le biais de mécanismes de supervision et de travail collaboratif.

6.4 Durée du projet

Les activités d'un projet de simulation nécessitent du temps. Ce temps est justifié par le fait que l'activité d'analyse et de conception est complexe, manipule de nombreuses données et contient une part d'irrationalité.

Les avantages de la durée sont la prise en compte de nombreuses informations et la possibilité de tester plusieurs scénarios au cours d'un projet. Ses inconvénients sont l'obligation de nombreuses mises à jour des objets produits, le coût de ces mises à jour, les informations qui se perdent faute d'être suffisamment ciblées, les oublis de décisions antérieures, etc.

La réduction de la durée globale d'un projet de simulation peut participer à un processus d'amélioration de la qualité. Les secteurs de l'automobile et de l'aéronautique nous en donnent l'exemple. La réduction du temps global n'est pas une diminution du temps utile, mais une réorganisation du temps. Elle passe en particulier par le parallélisme des tâches. Ce parallélisme

permet de prendre en compte des points de vue que le séquentiel ne permet pas. Bien que la concourance de l'activité d'ingénierie soit clairement posée, on ne peut oublier que, si les objectifs des acteurs d'un projet de simulation sont concourants, analyser et concevoir un système de production, les intérêts de ceux-ci sont souvent concurrents en terme de priorité financière, qualitative, etc.

L'appréhension du temps est un élément déterminant pour approcher le travail coopératif. En effet, selon la finesse temporelle avec lequel on analyse un projet de simulation, depuis la vue reculée et globale des grandes phases d'un projet jusqu'à la vue rapprochée et microscopique des relations, l'unité de temps donne une mesure différente aux protocoles de coopération. Par exemple, le chef de projet est à la croisée de ces multiples échelles de temps car il arbitre à la fois des contraintes de gestion appartenant au temps de l'entreprise-projet et des contraintes de réalisation appartenant au temps des équipes du projet.

7 Etat de l'art

7.1 Impact de la recherche sur le logiciel de simulation

En 1994 Mackulak (Mackulak et al., 1994) avait mené une étude en vue d'établir une liste de critères dont souhaitaient disposer les utilisateurs au sein du logiciel de simulation. Belattar (Belattar et al., 1997) considère que les résultats de cette étude sont très significatifs vu la notoriété des sociétés qui avaient participé à cette étude (IBM, Pritsker & Associates, CACI, Xerox, Aerospace, etc.) et dont certaines ont une grande expérience dans le développement des logiciels de simulation et d'autres comme utilisatrices de ce type de logiciel. Une autre étude récente a été menée par Tewoldeberhan (Tewoldeberhan et al., 2002) qui consiste à définir une liste de critères pour l'évaluation et la sélection des logiciels de simulation. La liste établie comporte 64 critères classés en 7 catégories :

- Modèle de développement et des entrées,
- Vendeur,
- Exécution,
- Animation,
- Testes et Efficacité,
- Sorties,
- Utilisateurs.

Ces critères sont surtout destinés aux équipes de développement qui en veillant à leur intégration dans le logiciel de simulation, garantiraient son succès auprès d'un large éventail d'utilisateurs. L'idéal serait donc de disposer d'un logiciel de simulation qui offrirait toutes ces possibilités à la fois. Cependant, il faut noter que dans la pratique, la prise en compte de trop de critères augmente les coûts de production du logiciel et complique sa réalisation. C'est pour cela qu'en général, les équipes de développement se contentent de satisfaire certains critères au détriment de certains autres.

Bien que l'objectif de produire des logiciels de simulation de qualité et conviviaux pour populariser la simulation est largement atteint (Belattar et al., 1997), nous estimons que la prise en compte des possibilités offertes par les technologies actuelles, en vue de rendre possible le travail en groupe qui est prépondérant dans tout projet de simulation, n'est pas encore pris en compte par les logiciels de simulation.

7.2 Simulation et TCAO

Les recherches dans le domaine de l'intégration des concepts du TCAO dans les environnements de modélisation et de simulation sont très limitées en nombre et en contenu. Les travaux publiés sur ce sujet discutent pour la plupart quelques aspects coopératifs d'un tel environnement sans donner des orientations sérieuses vers un environnement de modélisation et de simulation coopératif. Aujourd'hui, il existe un consensus sur les intérêts pratiques d'un tel environnement, mais des recherches plus approfondies sur ce sujet font toujours défaut. Cependant, nous devons souligner que la conception et la réalisation d'une application basée sur le travail coopératif pose un certain nombre de difficultés et des choix de conception qui doivent être faits à différents niveaux. En effet, dans ce type d'application, il devient nécessaire que des outils soient mis à la disposition des membres du groupe pour leur permettre de structurer leur travail ainsi que le fonctionnement de leur groupe de façon à assister véritablement le travail coopératif. Ceci ne peut être atteint que grâce à un environnement informatique plus riche que dans le cas d'une application centralisée et mono utilisateur.

En dépit des tendances actuelles visant à prendre en compte la dimension de groupe dans la conduite de projets en général, il existe peu de travaux de recherches similaires dans le domaine de la conduite de projets de simulation. La majorité des travaux ont consisté à développer des prototypes simples en vue de montrer les avantages de la simulation coopérative. Les principales tendances sont reportées dans ce qui suit :

Des éléments de support des collecticiels de simulation

Pour construire un environnement personnalisé de TCAO (de simulation par exemple), il est nécessaire d'utiliser des techniques et outils moins finalisés que les produits commercialisés. On est alors confronté au dilemme classique qui consiste à travailler avec des outils et des primitives de bas niveau ou utiliser des outils de plus haut niveau. Les premiers sont techniquement plus simples, en plus petit nombre, mais ils nécessitent un effort important de réalisation. Les seconds, techniquement plus complexes, sont généralement plus directement utilisables, mais aussi plus contraignants. Dans le contexte des collecticiels, deux possibilités sont donc offertes :

- utiliser directement des protocoles de communication standardisés tels que TCP-IP,
- utiliser des technologies de plus haut niveau proposant des mécanismes élaborés de programmation objet distribuée.

La première technique permet une liberté plus grande de conception et des performances potentiellement plus grandes que la deuxième. En revanche cette dernière permet un prototypage beaucoup plus rapide. Comme les protocoles de communications sont très largement connus et présentent un intérêt conceptuel limité, dans la suite nous allons présenter les dernières technologies de haut niveau proposées pour construire un collecticiel de simulation.

○ **Systèmes distribués et Simulation** : Un système informatique distribué est en général caractérisé par un ensemble de processus s'exécutant sur des ordinateurs connectés par un réseau de communication et par le biais duquel ces processus s'échangent de l'information. Les techniques de modélisation et d'implémentation qui ont été utilisées dans ce domaine ont été largement reprises dans le domaine de la simulation en vue de réduire principalement les temps d'exécution d'une simulation. Un modèle de simulation distribué peut donc être vu comme un ensemble de processus logiques s'exécutant de manière concurrente et modélisant chacun un composant physique du système réel modélisé ou processus physique. Les interactions entre les processus physiques sont modélisées par l'envoi de messages datés entre les processus logiques correspondants. Une autre approche de modélisation utilisant conjointement les concepts de l'A.O.O. et de l'informatique distribuée permet de voir un modèle de simulation comme un ensemble d'objets distribués sur un ensemble d'ordinateurs reliés par un réseau et communiquant par envoi de messages. Chaque objet sert dans ce cas à modéliser un composant du système réel. L'exécution d'un tel modèle de simulation consiste à invoquer les méthodes associées aux objets selon une séquence basée sur le temps. Du fait que les objets ne sont pas

liés dans un même programme exécutable, ceux-ci invoquent les méthodes par des envois de messages explicites. Parmi les projets les plus importants que nous avons recensé dans le domaine de la simulation distribuée nous avons retenu celui du développement d'un protocole pour la simulation distribuée et interactif ou D.I.S. (Distributed Interactive Simulation) initié par le département Américain de la défense (DoD). L'objectif principal du projet est l'interconnexion de différents simulateurs appartenant au DoD (simulateur de chars, d'avions, de détecteurs, etc.) en vue de simuler différents scénarios de combats. L'intégration des modèles environnementaux est aussi possible et permet de prendre en compte une certaine réalité dans laquelle évoluent les différentes entités. Il faut souligner que le protocole d'interconnexion qui a été défini dans le cadre de ce projet a été adopté comme un standard par IEEE. Un nouveau standard fut adopté en août 1996 sous le nom de HLA (High Level Architecture) conçu pour résoudre les problèmes et les limitations de D.I.S (problèmes d'évolutivité, limitation dans la gestion de plusieurs utilisateurs, surutilisation de la bande passante).

- **Web et Simulation** (Belattar et Korichi, 2001b): Aujourd'hui, on ne peut pas étudier le domaine du travail coopératif sans parler du World Wide Web. En effet, cet ensemble de réseaux d'information utilisant l'internet a sans doute été un des déclencheurs de l'engouement des constructeurs de logiciel pour le groupware. Grâce à lui, les utilisateurs ont pu commencer à partager une quantité colossale d'information, en organisant leurs communications autour de documents hypermédia (documents interconnectés utilisant textes, images et sons) rédigés avec un langage normalisé (HTML). En mettant à disposition des bases de données importantes, ils ont ainsi amorcé une forme de coopération implicite qui reste malheureusement généralement inconsciente. Durant ces dernières années, seule l'utilisation conjointe de la messagerie, et l'utilisation de formulaires permettaient d'aller un peu plus loin dans la conscience de la coopération. Depuis peu, le langage JAVA permet d'incorporer aux pages HTML du Web des petits programmes (applets). Ce ne sont plus seulement des informations qui sont transmises par le Web, mais aussi des traitements. Depuis ces dernières années, de nombreux travaux ont été publiés dans des conférences dédiées spécialement au thème de la simulation sur le Web. Plusieurs approches d'utilisation des techniques de simulation sur le Web ont été proposées. L'une d'elles consiste à soumettre le modèle de simulation depuis un navigateur vers un serveur en utilisant des formulaires dans des pages HTML. Le serveur traduit le contenu du formulaire en utilisant des scripts CGI en un programme qui sera exécuté sur le serveur par un Logiciel de Simulation cible (SIMAN, GPSS, QNAP, GPSS, etc.). Les résultats de la simulation sont transférés au client sous forme de page HTML. Une autre approche consiste à implémenter le modèle de simulation sous forme d'applet en utilisant le langage Java et qui sera exécuté du côté du client plutôt que sur le serveur. Dans le même contexte, beaucoup d'autres travaux sont

consacrés à la construction des modèles de simulation en utilisant le langage Java. Nous pouvons citer comme exemples : SIMJAVA, SILK.

Réalisations existantes

A notre connaissance, les travaux qui touchent les aspects coopératifs dans les environnements de simulation sont rares. Nous pensons que cela est dû d'une part au fait que le domaine du TCAO est très jeune et d'autre part qu'il n'y a pas beaucoup de groupes de recherches qui ont traité ce sujet.

Dans la suite nous présentons quelques réalisations permettent le travail coopératif dans le domaine de simulation que nous estimons intéressants.

- **NetMeeting & Simulation** (Taylor, 2000) : NetMeeting de MicroSoft a été utilisé dans plusieurs travaux afin d'élargir l'utilisation de la modélisation et simulation coopérative. Par exemple, Taylor a essayé de présenter le concept de la modélisation et simulation collaborative en utilisant NetMeeting. Cette approche est basée sur le concept de chatting, de tableau blanc partagé et de partage d'applications (outil de simulation). L'auteur recommandé la solution comme une amélioration du processus de modélisation et de simulation.
- **IVivist, NetMeeting & Simulation** (Klave et Stackenland, 2003) : Klave & Stackenland ont développés un prototype détaillé de modélisation et de simulation collaborative pour prouver les capacités de la collaboration à distance de la modélisation et de la simulation. Le prototype est basé sur l'outil de simulation Arena, la plateforme de collaboration IVisit pour la communication visuelle, et NetMeeting pour le partage de l'outil de modélisation et de simulation. L'auteur a eu l'honneur d'assister à la démonstration du prototype avec des participants éloignés géographiquement (entre la Suède et Singapour). Ce qui a confirmé les intérêts pratiques de la modélisation et la simulation collaboratives à très longues distances.
- **TangoSim** (Beca et al., 1997) : TANGOsims est une extension du système TANGO orientée vers les applications de simulation. Le produit original TANGO, appelé aussi système de collaboration, est un environnement de collaboration sur le WEB. Il est généralisable, flexible et extensible. Basé sur le travail collaboratif, Il permet la création de diverses zones d'information partagées et offre un support de haute qualité pour la visualisation et l'interaction avec l'utilisateur. De plus, pour différents utilisateurs connectés, il offre un support pour des

vues indépendantes et coordonnées. TANGO vise à étendre le paradigme du WEB au domaine du TCAO et bien au-delà du concept de chatting, de tableau blanc partagé et de partage d'applications génériques simples. L'adaptation de l'environnement TANGO aux applications de simulation (TANGOSim) est bâtie autour d'un simulateur à événements discrets "multi-threads" qui implémente la notion du temps virtuel et qui peut être piloté par un langage de scripts. Il peut être contrôlé interactivement par un utilisateur par l'intermédiaire d'un contrôleur de simulation. Le simulateur et le contrôleur sont respectivement implémentés sous forme d'application et d'applet Java. Le contrôleur de simulation peut générer des messages pour toute application compatible avec le système TANGO en vue de créer et contrôler des sessions et réaliser les scénarios dans lesquels la ligne de conduite dépend de la contribution de l'utilisateur. TANGOSim supporte aussi le filtrage des messages et l'enchaînement des messages entre différentes sessions d'applications. Malheureusement ce système n'a pas été utilisé ou développé davantage.

- **GroupSim** (Wladimir et al., 2004) : C'est un environnement de collaboration destiné à assister, avec les outils de simulation, le processus de construction et d'exploitation d'un modèle de simulation à événement discret en utilisant le Web comme plate-forme. Cet environnement est basé sur un système informatique collaboratif et exploite les concepts de la modélisation distribuée avec la génération automatique de programme et le contrôle distribué de l'expérimentation. Wladimir, Hirata et Edgar présentent quelques exemples pour illustrer l'utilisation de l'environnement et discutent quelques issues liées aux environnements de collaboration tels que la gestion des conflits d'accès, le contrôle d'accès, la conscience de groupe ainsi que la performance.
- **COVISE** : (**CO**llaborative **VI**sualization and **S**imulation **E**nvironment) (Lang et al, 1995) est un environnement distribué et extensible. Il intègre les fonctionnalités de simulation, post-traitement, multimédia et de visualisation basées sur le travail coopératif. COVISE a été conçu de manière à permettre l'incorporation des différentes infrastructures de haute performance telles que les calculateurs parallèles, calculateurs vectoriels et les réseaux rapides. Ainsi, COVISE peut tirer profit des caractéristiques variées de ces infrastructures. La conception ouverte de COVISE permet l'extension facile de son architecture et d'intégrer de nouveaux programmes industriels ou de recherches en simulation sous forme de nouveaux modules au sein de son architecture. Son développement a été lancé dans le projet PAGEIN (Pilot Applications in a Gigabit European Integrated Network). L'architecture de base ainsi que plusieurs modules ont été développés au centre de calcul de l'Université de Stuttgart en Allemagne depuis 1993. COVISE est utilisé dans des grands projets de coopération comme le projet ESPRIT et

ADONNIS (A Demonstration of New Networking Integrated Services) pour permettre la direction et le contrôle directs de la simulation continue.

8 Conclusion

Dans ce chapitre, nous avons examiné le contexte d'un collectif de simulation par la présentation de la méthode de conduite d'un projet de simulation et la justification de l'adéquation des concepts du TCAO avec cette méthode. Nous avons ensuite présenté un état de l'Art des travaux portant sur la coopération et la collaboration dans le domaine de la simulation. Cet état de l'Art nous permet d'une part, de justifier et de motiver nos travaux, et d'autre part, de situer nos choix privilégiés dans les phases de conception et d'implémentation.

L'étape suivante consistera à examiner les travaux de recherche issus du domaine de travail coopératif assisté par ordinateur en terme d'outils de conception et de développement logiciel des applications de groupe. Ceci en vue de choisir celle qui conviendrait le mieux au problème traité dans ce travail de recherche.

Chapitre n° 3

Les Collecticiels

III Les Collecticiels

1 Introduction

L'émergence des collecticiels est depuis les années 80 lente mais progressive. Elle connaît depuis peu un essor important avec l'avènement des réseaux informatiques et de la téléphonie mobile. Le World Wide Web n'est pas en reste, puisqu'il participe à cette démocratisation progressive de l'accès aux réseaux informatiques. Ainsi, les collecticiels ne sont plus confinés aux laboratoires de recherche et ne sont plus strictement réservés aux informaticiens. Ils sont de plus en plus acceptés et accaparés par le grand public. Parmi les exemples de collecticiels quotidiennement utilisés aujourd'hui, on cite : les jeux en réseau, les forums de discussion (chat et newsgroup), les échanges de courriers électroniques, les échanges de fichiers, les communautés virtuelles mais réelles comme les communautés de développement logiciel Open Source, la vidéoconférence...Etc. Cette situation a poussé les chercheurs dans le domaine du TCAO à investir dans les développements des outils de mise en œuvre des collecticiels, que ce soient des outils de conception ou de réalisation.

Pour mieux maîtriser les méthodes de conception et de réalisation des collecticiels, et en vue de les appliquer dans la phase de conception de notre collecticiel, ce second chapitre sera consacré essentiellement aux collecticiels. Celui-ci comportera les notions fondamentales liées au domaine du TCAO, un état de l'art des méthodologies de conception et des outils informatiques déjà disponibles. Cette étude va nous permettre de mieux cerner le domaine des collecticiels pour atteindre notre objectif qui est la conception et la réalisation d'un collecticiel de modélisation et de simulation.

2 Terminologies

TCAO

Le terme anglo-saxon CSCW, désignant "Computer Supported Cooperative Work", est employé dans la littérature la première fois par deux chercheurs : Irene Greif et Paul Cashman en 1984 pour définir l'ensemble des systèmes informatiques qui facilitent la coopération d'individus autour d'une tâche commune (Payet, 2003). La traduction française de ce terme est "**T**rabail **C**oopératif **A**ssisté par **O**rdinateur" (**TCAO**). Depuis le sigle TCAO est utilisé pour nommer le domaine de recherche qui traite de la conception de systèmes d'application pour une catégorie de travail spécifique : le travail coopératif dans toutes ses formes. L'autre terme anglo-saxon souvent utilisé en substitution de CSCW est groupware. Ce terme, initialement employé dans le langage courant des scientifiques américains, a vite connu une utilisation plus universelle dans la littérature informatique. En 1978, Peter et Trudy Johnon-Lenz, pionniers dans le travail coopératif, introduisirent le terme groupware pour définir une activité de groupe intentionnelle augmentée d'un support logiciel permettant sa réalisation. "**C**ollecticiel" est la traduction officielle préconisée par l'Association Française pour la Cybernétique Économique et Technique (ASTI) pour désigner groupware. On trouve aussi très souvent le terme d'«application collaborative» pour traduire le mot «groupware». Tarpin (Tarpin-Bernard, 1997b) considère que le qualificatif «application» est inapproprié compte tenu des exigences en terme d'ouverture et d'évolutivité mises en évidence dans la recherche en TCAO pour ce type de programme. Le mot «application» a une connotation de programme dédié à une tâche précise, aussi il est préférable d'utiliser le mot «système» qui souligne bien plus le mode de travail (la collaboration) qu'une activité particulière.

Collecticiel

Le TCAO devrait être considéré comme une tentative pour comprendre la nature et les caractéristiques du travail coopératif, avec comme objectif la conception d'une technologie adéquate. Cette technologie, lorsqu'elle prend la forme d'un environnement informatique destiné à assister un groupe de personnes engagées dans une tâche commune, s'appelle un collecticiel (Payet, 2003).

Par définition les collecticiels sont considérés comme des outils permettant à plusieurs groupes d'individus, géographiquement distribués, de travailler ensemble avec l'aide d'un environnement informatique. L'importance de ce type d'outil est ainsi recentrée sur les utilisateurs et leur contexte de travail respectif. Un outil collaboratif doit permettre à un groupe d'individus de travailler ensemble, sur les mêmes données. Les individus peuvent se trouver dans un même lieu ou être

éloignés de plusieurs milliers de kilomètres. Ils peuvent travailler en temps réel, de façon synchrone ou de façon asynchrone. Ils peuvent utiliser un parc matériel varié et avoir des systèmes d'exploitation différents les uns des autres.

De nombreuses définitions ont été proposées pour caractériser un collecticiel dont nous citons la plus courante, celle de C. Ellis (Laurillau, 2002) :

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

“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 (Laurillau, 2002))

Coopération, travail coopératif et tâches communes

La définition littéraire du terme coopérer est Agir conjointement avec quelqu'un. La coopération en informatique est une instanciation de cette définition en utilisant des moyens logiciels et matériels. Elle désigne une activité informatique menée par un groupe d'utilisateurs dans un espace informatique commun.

Dans ce contexte, le travail coopératif désigne un travail faisant intervenir plusieurs utilisateurs sur des ressources informatiques communes.

Le terme tâche peut être employé pour définir une partie d'un travail coopératif. Elle peut résulter d'un découpage du travail en plusieurs parties, en fonction de l'un des critères suivants : la spécificité (e.g. édition, dessin, ...), les attributions des participants, le partage d'un même sous-ensemble de ressource, etc. Souvent, les termes tâche et travail sont employés pour désigner la même chose.

Acteur

Les utilisateurs d'un collecticiel sont souvent appelés les acteurs de la coopération. Le terme participant est également employé pour distinguer entre un utilisateur dans un environnement non coopératif et un intervenant dans un collecticiel.

Groupe

Le terme groupe est employé dans plusieurs domaines tels que les sciences sociales, la médecine ou les mathématiques. Il désigne un ensemble d'individus ou d'entités partageant des caractéristiques communes. Ces caractéristiques peuvent être statiques ou dynamiques, et définissent un comportement commun. Dans le contexte du collecticiel, le groupe est employé pour désigner un ensemble d'utilisateurs impliqués dans un travail commun.

Contexte partagé

Un contexte partagé désigne un ensemble de ressources accessibles par un groupe d'utilisateurs. Ces ressources peuvent être des documents partagés, des informations sur l'état d'exécution d'un processus, etc.

Rôle et vue

Le terme rôle est utilisé pour définir les attributions d'un acteur dans un collecticiel. Le terme vue est employé pour définir une visuelle (La perception de quelque chose à travers la vue) d'un ensemble d'informations du contexte partagé. La vue peut être attachée à un rôle. La différence essentielle entre vue et rôle peut être définie de la manière suivante :

- le rôle permet de définir les actions que peut mener l'acteur sur le contexte partagé,
- la vue permet de définir comment l'acteur perçoit les actions réalisées par d'autres acteurs sur le contexte partagé,
- enfin, différentes vues peuvent représenter la même information, alors que, deux rôles distincts confèrent nécessairement des attributions différentes.

Par exemple, un participant peut avoir un rôle de président dans une réunion de travail lui conférant le droit d'accepter ou de refuser les requêtes d'adhésion de nouveaux participants. Un utilisateur peut également avoir un rôle de rédacteur ou de simple lecteur sur un document partagé.

3 Les fonctionnalités d'un collecticiel

Les collecticiels proposent de compléter les fonctionnalités habituelles des logiciels par des fonctions qui permettent à un collectif de participants de communiquer, de co-produire (c'est-à-dire d'agir ensemble sur des données partagées), et de se coordonner. Pour structurer l'analyse des collecticiels, et proposer un moyen de classification des nombreux outils permettant d'atteindre les objectifs du TCAO, certains auteurs répartissent les fonctionnalités des outils coopératifs dans le trèfle fonctionnel. Représentation qu'ils ont nommée « The clover model » (Payet, 2003), celle-ci est aujourd'hui communément référencée sous le nom de « Trèfle du collecticiel ».

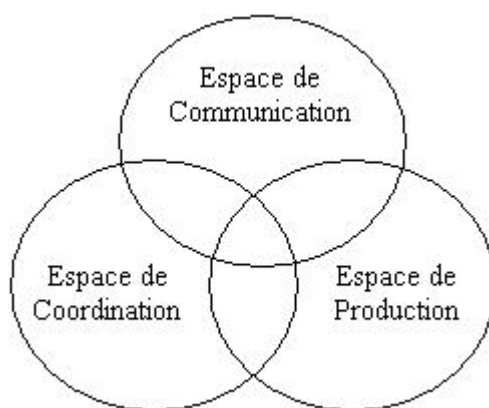


Figure 13 : Trèfle du Collecticiel (Tarpin-Bernard, 1997b)

Les trois espaces de ce modèle sont définis ainsi :

- L'espace de production comprend les artefacts partagés qui sont manipulés de façon collaborative pour réaliser les différentes tâches. Autrement dit toutes les fonctionnalités de construction et d'accès aux données partagées. On y retrouve l'ensemble des outils liés au domaine d'activité.
- L'espace de coordination codifie les protocoles qui gouvernent la façon dont les tâches sont gérées par les groupes d'utilisateurs. Ces protocoles peuvent être purement sociaux, comme par exemple la règle sociale qui spécifie que seulement une personne doit parler à la fois, ou peuvent être formellement spécifiés à travers un système de type workflow.
- L'espace de communication concerne les communications entre individus. Souvent, le vocable communication est utilisé de façon ambiguë. Franck TARPIN (Tarpin-Bernard, 1997b) préfère en effet limiter le mot *communication* à l'échange de données informatiques, et parler de *conversation* lorsqu'il s'agit d'un échange interpersonnel de signaux, numériques ou analogiques, ne pouvant

être interprétés par la machine. Les progrès des technologies réseaux et multimédia ont permis de voir apparaître une multitude de modes de conversation. Par exemple, le courrier électronique et les médiaspace (systèmes conçus pour supporter la communication synchrone par ordinateur interposé).

Assister l'interaction dans ces trois espaces fonctionnels constitue la base de tout support de travail coopératif. Cependant, lorsque le groupe s'agrandit et s'organise, (Tarpin-Bernard, 1997b) souligne la nécessité de prise en compte de trois nouveaux enjeux, à savoir :

- Favoriser la cohésion du groupe : Pour renforcer la cohésion du groupe, il faut aider les participants à se connaître. Chacun doit pouvoir obtenir des informations sociales et organisationnelles (statuts, rôles) sur les autres membres. De plus, des informations globales partagées portant sur la tâche et les activités actuelles du groupe renforcent l'appartenance objective et subjective au groupe. Enfin, pour entretenir la cohésion de groupe, il faut pouvoir imprimer des rythmes de travail au groupe, et favoriser la prise de décision en commun.
- Favoriser l'implication individuelle : L'homme montre une résistance naturelle au travail en groupe, il faut donc éviter toute cassure entre le travail individuel et le travail pour le groupe. On a donc tout intérêt à permettre l'intégration dans le travail du groupe des résultats d'un travail individuel. Un utilisateur apprécie également de pouvoir utiliser ses outils personnels pour participer directement au travail du groupe. Les outils collaboratifs doivent pouvoir être adaptés au profil et au rôle de chacun, afin que les participants ne se sentent pas contraints pour participer au groupe. Réciproquement, il faut permettre d'utiliser les outils de travail et les ressources du groupe pour travailler de façon isolée. Enfin, la résistance au travail en groupe provient aussi de la peur du participant d'être 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.
- Faciliter l'organisation du groupe : Lorsqu'un groupe devient une organisation, tous ses membres ne participent pas à toutes les activités du groupe. Il se divise alors en équipes de travail. Il faut donc permettre la constitution des informations organisationnelles nécessaires à la cohésion du groupe. Les équipes et leurs constitutions sont très variables et changeantes. Elles doivent pouvoir construire des espaces de collaboration qui s'enrichissent au fur et à mesure que l'organisation se développe.

D'une façon plus générale, ces trois nouveaux enjeux ont pour fondement le soutien du moral du groupe. Cette problématique est classique en sociologie. Ainsi cet objectif est lié au quadruple détermination suivante (Tarpin-Bernard, 1997b) :

- La conscience d'être ensemble et de coopérer,
- Le sentiment d'avoir un objectif,
- La possibilité d'observer un progrès dans la marche vers l'objectif,
- Le fait que chaque membre soit responsable de tâches spécifiques significatives qui sont nécessaires à l'accomplissement de l'objectif.

Des travaux ont été menés par Hoogdtoel (Hoogdtoel, 1996) en vue de dégager une liste de fonctionnalités à intégrer dans un environnement de TCAO. Cette étude a donné lieu à une liste composée de quarante six (46) fonctionnalités que nous rapportons dans le tableau suivant :

Fonctionnalités	
1.	Utiliser différents moyens de communication (vidéo, son, texte)
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
7.	Désigner les objets de la conversation
8.	Partager des espaces d'action (= supporter la coopération fortement couplée)
9.	Gérer les risques de conflits d'accès aux espaces d'action communs
10.	Structurer la conversation
11.	Informations sociales et organisationnelles
12.	Communication informelle
13.	Attribuer des rôles fonctionnels
14.	Connaître les rôles fonctionnels
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 dans les différentes situations de coopération
20.	Pouvoir faire évoluer les rôles
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
24.	Utiliser des outils personnels pour participer au travail du groupe
25.	Intégrer des activités non informatisées
26.	Adapter les outils à chaque individu
27.	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

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'espace d'information commun en situation de coopération isolée
33.	Accéder à l'espace d'information commun en situation de coopération séparée
34.	Accéder à l'espace d'information commun en situation de coopération faiblement couplée
35.	Accéder à l'espace d'information commun en situation de coopération fortement couplée
36.	Fournir des services de recherche des ressources
37.	Structurer l'espace d'information commun en fonction de l'organisation
38.	Structurer l'espace d'information commun en fonction des tâches
39.	Répertorier les membres
40.	Gérer la dynamique des rôles organisationnels
41.	Former des équipes
42.	Créer des espaces de coopération persistants
43.	Configurer les modes de coopération des activités de groupe
44.	Configurer les espaces de coopération
45.	Organiser l'univers des espaces de coopération
46.	Définir et intégrer de nouvelles activités de groupe

Tableau 4 : Liste de fonctionnalités requises dans un collecticiel

Cette liste sera certainement d'une aide efficace dans le processus de conception et d'implémentation de tout environnement supportant le travail de groupe et plus précisément l'environnement de simulation coopératif que nous envisageons. Il faut souligner à ce niveau, qu'il n'est pas impératif d'intégrer toutes ces fonctionnalités en même temps tout en sachant qu'un environnement qui les intégrerait toutes serait l'idéal. Nous essaierons de mettre en évidence l'adéquation de l'environnement que nous proposons avec les fonctionnalités énumérées ci-dessus (voir chapitre n° 4)

4 Paramètres des collecticiels

4.1 Modes d'interaction : collecticiels synchrones/asynchrones

Les collecticiels offrent plusieurs possibilités pour échanger des informations entre les participants. Un collecticiel est dit synchrone si les informations sont échangées selon un mode temps réel. Dans ce cas, les acteurs sont avertis en temps réel des actions menées sur les ressources partagées.

Le mode d'interaction synchrone nécessite que les membres du groupe soient présents en même temps pour effectuer le travail coopératif. Les systèmes de téléconférence sont des exemples représentatifs de ce mode de fonctionnement. Ce mode permet un partage immédiat des informations.

Les collecticiels asynchrones, au contraire, correspondent aux applications coopératives dont le support de communication est asynchrone, de type courrier électronique. Le mode d'interaction asynchrone ne nécessite pas la coprésence des différents participants. Ce mode est fréquemment employé dans l'édition coopérative de documents. Le mode asynchrone permet un partage séquentiel des informations.

La classification des collecticiels en fonction du mode d'interaction ne recouvre pas complètement l'ensemble des tâches coopératives. L'édition coopérative représente un exemple de collecticiels qui peuvent être à la fois synchrone et asynchrone.

4.2 Le mode de fonctionnement WYSIWIS

Le terme WYSIWIS signifie en français "ce que tu vois est ce que je vois". Ce terme est utilisé pour définir le fonctionnement d'un collecticiel fournissant une vue cohérente du contexte partagé, à tout moment de la coopération, pour tous les acteurs.

La différence essentielle entre les termes : synchrone et WYSIWIS réside dans le fait que le mode synchrone définit la communication temps réel des actions, alors que le mode WYSIWIS caractérise la perception visuelle des actions communiquées (en mode synchrone).

Les relations logiques qui existent entre ces deux concepts sont les suivantes :

- un mode de fonctionnement WYSIWIS implique nécessairement un mode d'interaction synchrone,
- alors qu'un mode d'interaction synchrone ne suffit pas pour définir un mode de fonctionnement WYSIWIS

Le WYSIWIS strict représente l'application extrême du mode de fonctionnement WYSIWIS. Il consiste à avoir une vue identique des informations partagées sur tous les terminaux des participants. La mise en œuvre de ce fonctionnement nécessite un support matériel identique pour tous les acteurs, le cas échéant, des mécanismes de conversion capables d'adapter la taille des fenêtres à celles des écrans. Pour relaxer le WYSIWIS strict, il existe plusieurs approches. Ces approches sont fondées sur les aspects visuel, temporel, taille (taille du groupe d'acteurs). La relaxation temporelle poussée peut conduire vers un mode de fonctionnement asynchrone.

4.3 Cohérence

Dans les collecticiels synchrones, le mode WYSIWIS strict définit une cohérence forte du contexte partagé. Selon ce mode, tous les acteurs ont une vue identique des données partagées.

La cohérence faible peut être engendrée par le mode WYSIWIS relaxé selon l'aspect temps. Une telle relaxation donne dans le cas extrême, où aucune hypothèse n'est faite sur le mode d'interaction, un fonctionnement asynchrone.

Cependant, le mode WYSIWIS relaxé selon l'aspect visuel n'affecte pas la cohérence du contexte partagé. Même si les acteurs n'ont pas la même vue des informations, l'état de ces informations est identique pour tous.

4.4 Granularité

La granularité est une caractéristique spatiale qui, à l'opposé, du mode d'interaction, est fréquemment associée aux droits d'accès aux données attribués aux utilisateurs du collecticiel. La granularité définit l'unité d'information qui peut être accédée simultanément par plusieurs utilisateurs.

Une granularité fine associée à un mode d'interaction synchrone définit un travail fortement couplé. Dans le cas contraire, une granularité forte associée à un mode asynchrone définit un travail faiblement couplé.

Par exemple, un éditeur de texte permettant l'accès simultané à un même mot permet une granularité plus fine qu'un éditeur de texte ne permettant l'accès au document d'un utilisateur à la fois.

Dans cas des collecticiels qui utilisent des politiques à jeton, si celui-ci est attribué pour accéder tout un document, nous sommes dans le cas d'une granularité est forte.

4.5 Droits d'accès, droit de parole, politiques à jetons

Les droits d'accès associés à un utilisateur sont en partie constitués des attributions qui lui sont conférées par son rôle, et en partie par les droits qui peuvent lui être confiés pendant le déroulement

du travail coopératif. Ces droits, dynamiquement attribués, sont communément appelés droits de parole.

Dans les collecticiels, le droit de parole est échangé entre les participants à l'aide de politiques à jeton. Il s'agit de différents protocoles qui permettent aux participants de se communiquer une information de contrôle désignant un jeton logiciel.

En général, les politiques utilisent un seul jeton. Ce procédé permet de garantir un accès exclusif aux informations partagées. Toutefois, dans certaines applications, plusieurs jetons peuvent coexister, c'est souvent le cas des applications de jeux virtuels où m utilisateurs parmi n ($m \leq n$) peuvent modifier le contexte partagé.

Les politiques de passage du jeton sont regroupées en deux classes. Celles-ci sont respectivement fondées sur :

- la désignation explicite. Un acteur se charge de passer explicitement le jeton à un autre membre du groupe. Celui-ci peut être l'ancien détenteur du jeton ou un acteur auquel revient la responsabilité d'organiser le travail et de coordonner les interventions des participants,
- le passage implicite. Dans ce cas les acteurs n'interviennent pas explicitement dans l'affectation du jeton. L'application se charge de désigner le détenteur du jeton en fonction de l'ordre d'arrivée des demandes pour l'acquisition du jeton (e.g. ordre FIFO), ou en fonction d'une quelconque priorité instaurée entre les participants.

4.6 Accès libres, conflits d'accès et synchronisation

L'adoption du mode d'accès libre peut engendrer des conflits et des incohérences de l'espace partagé. Un exemple simple d'une telle situation est observé lorsque deux opérations conflictuelles (opérations d'écriture): op_1 , op_2 , sont effectuées par deux acteurs sur un même objet (l'objet partagé est dupliqué sur tous les sites participants). Si l'exécution de ces deux opérations n'est pas faite selon le même ordre sur tous les sites, l'état de l'objet partagé sera incohérent.

Afin de garantir la cohérence dans les collecticiels synchrones, le système doit synchroniser les actions des différents participants sur tous les sites afin de maintenir un état cohérent pour toutes les données.

4.7 Gestion dynamique de groupes : protocoles de connexion/déconnexion de participants dans un collecticiel

Un service de gestion de groupe dans collecticiel fournit principalement des protocoles permettant : la connexion de nouveaux membres et la déconnexion d'anciens membres.

La participation dynamique de participants figure parmi les fonctions spécifiques fournies par un collecticiel synchrone. Cette fonction marque le passage d'un utilisateur d'un environnement de travail privé à un environnement de travail coopératif.

Afin de garantir la flexibilité d'utilisation, le collecticiel doit permettre à des retardataires de se connecter pendant le déroulement du travail. La complexité de cette opération consiste à fournir au nouvel arrivant une vue "cohérente" du contexte partagé et à informer les membres du collecticiel de l'arrivée d'un nouveau membre.

De même, le collecticiel doit gérer le départ de certains membres qu'il soit volontaire ou résultant d'une panne quelconque du réseau ou des sites.

En résumé, deux contraintes régissent l'exécution des fonctions de participation et de déconnexion. Elles sont respectivement spatiales et temporelles :

- cohérence du contexte partagé : pour les collecticiels adoptant un mode de fonctionnement WYSIWIS, à l'issue de la phase de connexion, le nouvel arrivant doit avoir une vue identique du contexte partagé. Dans ce cas, le protocole garantit une cohérence forte,
- temps de réponse : le traitement des demandes de connexion ou de déconnexion doit être asynchrone. le protocole de connexion doit garantir des temps de réponse faibles aussi bien pour le demandeur de la connexion que pour les acteurs du collecticiel.

La cohérence du contexte n'est pas absolue, elle est liée au rôle, voire à la vue, qui sont conférés à l'utilisateur au moment de son adhésion au groupe. Par exemple, si un utilisateur n'est autorisé qu'à partager une application particulière du collecticiel en mode WYSIWIS, le service de gestion de groupe ne lui fournit que le contexte de l'application en question.

La fonction de déconnexion marque la sortie d'un acteur du groupe de participants. Le protocole de déconnexion doit garantir un fonctionnement normal du collecticiel pendant cette phase. En général, le protocole doit s'assurer que le participant n'a pas de jeton, ou qu'il n'occupe pas de rôle pouvant entraver le déroulement du travail (par exemple, le rôle de président doit être cédé avant de quitter la session).

Les opérations de connexion/déconnexion dynamiques sont accomplies par des protocoles qui sont souvent choisis en fonction du type du travail coopératif. Dans certains collecticiels, la connexion d'un utilisateur peut être soumise à l'accord d'un président. Dans d'autres applications telles que les téléconférences simulant par exemple un séminaire grand public, la connexion d'un nouvel arrivant est complètement transparente aux acteurs du collecticiel.

4.8 Session

Une session permet d'associer : un groupe d'acteurs, un contexte partagé, des rôles, des protocoles d'accès, et des protocoles de connexion/déconnexion, afin de définir selon la dimension espace un travail coopératif. La définition complète du travail nécessite la donnée du mode d'interaction adopté. Souvent, le terme session est employé pour définir une durée d'exécution d'un collecticiel synchrone.

4.9 Les concepts IHM

Plusieurs concepts hérités du domaine interface homme-machine sont également utilisés dans le domaine du collecticiel. Des termes comme l'observabilité et l'honnêteté décrivant les caractéristiques des interfaces utilisateur dans les collecticiels :

- l'observabilité est la capacité du système à rendre perceptibles à l'utilisateur les variations des données faisant l'objet de la coopération,
- l'honnêteté : le système est honnête si le rendu de l'état observable des données est conforme à l'état interne, et si la forme de ce rendu conduit l'utilisateur à interpréter correctement cet état (c'est à dire, le résultat de cette interprétation),
- la pro-activité des retours d'information caractérise un comportement qui empêche l'utilisateur d'accomplir une action interdite. Par exemple, un élément de menu en grisé

indique à l'utilisateur qu'il est inopérant (caractère observable) et, en même temps, n'active pas de fonction système qui conduirait à une erreur (il est pro-actif).

- o la conformité du temps de réponse mesure la capacité du système à réagir dans un laps de temps en accord avec l'attente de l'utilisateur.

5 Classifications des collecticiels

L'approche traditionnelle pour présenter les collecticiels consiste à utiliser la classification espace-temps d'Ellis (Ellis et al. 91). Suivant cette classification, les systèmes se différencient par la distance temporelle entre les utilisateurs et par leur distance spatiale. Les utilisateurs interagissent de manière synchrone s'ils manipulent simultanément les données partagées, ou de manière asynchrone s'ils les manipulent des moments différents. De même ils peuvent interagir en étant dans la même pièce ou dans des lieux différents. Stéphane Sire (Sire, 2000) propose une autre classification fonctionnelle basée sur le modèle trèfle fonctionnel que nous résumons ici.

5.1 Systèmes pour la communication

Les outils conçus pour faciliter l'échange de messages et la communication entre les personnes sont les outils de CMC (Computer Mediated Communication). Ces outils rapprochent les groupes de l'idéal de la communication totale. L'objectif de ces outils est de fournir aux groupes la possibilité de parvenir à une forme d'intelligence collective. C'est-à-dire que les résultats des processus de communication dans le groupe sont supérieurs aux résultats que chaque individu aurait pu obtenir individuellement.

Pour y parvenir, les outils de CMC doivent : rendre possible les communications dans le groupe permanente et offrir la possibilité d'être utilisés de manière synchrone ou asynchrone.

Cette deuxième caractéristique est le potentiel de facilitation sociale, c'est-à-dire la capacité des outils de CMC à favoriser la mise en communication spontanée des individus, à l'image des rencontres de couloir. Lorsque ces outils sont utilisés pour connecter les individus de manière permanente, ils créent des espaces virtuels qui acquièrent une existence propre, au même titre que les espaces physiques. Ces outils se distinguent en fonction de la nature des canaux sur lesquels ils reposent. Les médiaspace utilisent des canaux audiovisuels, les messageries et les forums de discussion des canaux textuels, les MUD (Multi-User Dungeons) au départ textuels reposent de plus en plus sur des environnements de réalité virtuelle.

Médiaspaces

Les médiaspace sont des installations qui relient des réseaux audio et vidéo à des réseaux informatiques. Contrairement à la vidéoconférence, les connexions vidéo des médiaspace sont disponibles en permanence. Dans un médiaspace, chacun possède un accès à un ensemble de caméras avec lesquelles il peut établir différents types de connexions : communication visiophonique, “coup d’œil”, ou encore mode continu. Le coup d’œil est une connexion vidéo très brève et unilatérale qui permet de vérifier la disponibilité d’un correspondant. Il est parfois accompagné d’un signal sonore pour prévenir la personne observée. Le mode continu est une connexion unilatérale permanente généralement employée en arrière-plan pour percevoir l’activité ou vérifier la présence d’autrui.

Messageries et forums de discussion

Les outils de dialogue textuel sont anciens et nombreux. Les outils de dialogue en ligne servent à dialoguer en mode synchrone, c’est-à-dire tous les utilisateurs interagissant de manière simultanée, comme l’IRC (Internet Relay Chat). Les messages sont adressés à un canal et sont reçus par tous les utilisateurs qui écoutent le canal. L’échange de messages textuels existe aussi sous la forme des commandes `write` ou `msend` sur UNIX pour envoyer des messages sur le terminal d’un autre utilisateur. La commande `talk` partage une fenêtre en deux moitiés où deux utilisateurs connectés peuvent s’échanger des messages en mode synchrone.

Le courrier électronique s’est imposé comme une réussite incontestable de l’informatique pour favoriser les activités de groupe. En effet, le courrier électronique est avantageux pour le solliciteur à qui il permet d’envoyer à tout moment ses requêtes de manière plus ou moins intrusive pour le destinataire. Néanmoins, il respecte l’équilibre entre les deux car l’effort de saisie demandé à l’expéditeur est supérieur à l’effort de consultation demandé au destinataire. De plus le destinataire garde le contrôle du moment auquel consulter le message. Le courrier électronique n’impose pas non plus de contenu aux messages ni de règles de diffusion, ce qui laisse une place aux pratiques sociales pour réguler leur diffusion et leur format.

Le courrier électronique s’utilise aussi avec des listes de diffusion pour envoyer le même message ou le même document à une liste de destinataires. Il est complété par les systèmes de messageries pour archiver des messages et des documents et les faire circuler sur un ensemble de serveurs. Ces archives sont accessibles à tout le monde à tout moment et sont structurées par

groupe d'intérêt. Les news couvrent ainsi l'ensemble de la planète à partir du réseau Internet. Ces systèmes d'échange de messages sont à la base de véritables communautés virtuelles.

5.2 Systèmes pour la production

La facette production domine lorsque les collecticiels servent à produire un artefact électronique plutôt qu'à communiquer autour de cet artefact. Les applications correspondantes sont les éditeurs partagés. Elles se distinguent entre elles par le type de données partagées, par la fréquence des mises-à-jour des données et par le couplage entre les vues des utilisateurs.

Les éditeurs partagés sont des applications qui permettent de produire en commun divers types de contenus: texte, comme, Duplex, ou Alliance, dessin, comme GroupDesign, etc. Les tableaux blancs partagés servent à éditer à plusieurs des dessins et des textes non structurés dans une fenêtre. Les auteurs éditent de manière synchrone ou asynchrone les mêmes documents. La notion de contenu s'étend à une multitude de types de données même en l'absence de produit final. Par extension tout système destiné à partager et à modifier un état à plusieurs peut s'envisager comme un éditeur partagé.

Les éditeurs partagés reposent sur des mécanismes de partage des données. Ces mécanismes s'appliquent à une quantité minimale d'information qui est mise à jour simultanément sur toutes les répliques: la granularité. Les techniques mises en œuvre pour la réalisation des éditeurs partagés sont les mêmes que pour les systèmes distribués: contrôle d'accès concurrent, gestion de la cohérence et réplication des données

Au niveau de l'interface homme-machine les éditeurs partagés doivent assurer la cohérence entre les vues des utilisateurs. Une vue correspond à une représentation des données partagées. Les vues de chaque participant sont plus ou moins semblables aux vues des autres participants. Cette ressemblance, ou ce couplage entre les vues peut-être flexible et varier d'une indépendance totale à une complète similitude. Le couplage maximal est obtenu en appliquant strictement la notion de WYWIWIS. Tous les éléments de l'interaction sont alors partagés et chaque utilisateur voit exactement ce que les autres voient. Les télépointeurs servent à reproduire à distance les pointeurs des autres utilisateurs.

5.3 Systèmes pour la coordination

Les outils de CMC sont orientés vers la communication. Les éditeurs partagés sont orientés vers la production. Ils correspondent dans le requis de la production totale à la gestion des dépendances de partage de ressource. Une troisième classe d'application correspond également au requis de la production totale. Cette classe est orientée vers la gestion des dépendances de producteur à consommateur. Il s'agit des systèmes basés sur le traitement automatique des messages et des documents d'un groupe. Ces systèmes privilégient la facette coordination définie par le trèfle. Nous distinguons les coordinateurs et les applications workflow

Coordinateurs

Les coordinateurs ont pour but de faciliter la coordination au sein d'un groupe en aidant les utilisateurs à organiser leurs tâches. Il s'agit par exemple des agendas de groupe comme l'application commerciale TeamAgenda. D'autres coordinateurs servent à aider les individus à traiter des affaires autrefois réglées par des conversations téléphoniques ou en face-à-face en les remplaçant par des échanges de messages électroniques. Dès le milieu des années 80, ce type de coordinateur est apparu pour aider les individus à mémoriser les messages à traiter en priorité, en fonction de l'état des conversations, terminées ou en attente de réponse. Ils permettent aussi de représenter l'historique des conversations. La réalisation de coordinateurs de messages est limitée par la difficulté à faire comprendre des messages humains à un ordinateur. Cette limitation est contournée si l'utilisateur indique explicitement à l'ordinateur la catégorie de message qu'il envoie. L'ordinateur est alors capable de traiter les messages.

D'autres types de coordinateurs sont basés sur la représentation graphique des conversations conservées dans des bases de données. Ces systèmes mémorisent les traces des processus de décisions. Oval est un outil de recherche pour construire des interfaces à des systèmes d'échange de messages structurés en réseaux d'argumentation, c'est-à-dire qui se rapportent les uns aux autres en tant qu'argument ou contre-argument. Les réseaux d'argumentation sont des réseaux hypertextes. Dans le cadre de la gestion d'un projet, les réseaux d'argumentation mémorisent les étapes importantes, justifient les choix effectués ou servent à détecter les erreurs. Dans le commerce les logiciels de mémoire organisationnelle servent à construire des réseaux d'argumentation, comme QuestMap.

Workflow

La frontière est étroite entre les coordinateurs et les applications workflow qui servent à coordonner des processus au sein d'une entreprise. Ceci est justifié par l'inexistence d'une définition unique pour les applications workflow. Les applications workflow automatisent la gestion des flux d'information suivant les spécifications d'un processus donné. Les tâches de traitement de l'information passent d'une personne à une autre suivant un circuit conditionnel bien défini. (...) L'application présente à l'utilisateur les informations nécessaires pour effectuer sa tâche, avant que le processus ne suive son cours vers l'étape suivante.

La mise en œuvre des applications workflow passe parfois par un remodelage complet de l'organisation du travail autour de ses processus. Ce remodelage est un "Business Process Reengineering" (BPR). Le BPR est une opportunité pour réduire les coûts et améliorer l'efficacité du travail. La mise en œuvre des applications workflow nécessite une modélisation de processus qui doit expliciter le fonctionnement d'un groupe. Cette modélisation est souvent basée sur la représentation des documents, de leur circulation et des modifications apportées aux documents au cours des différentes étapes de leur diffusion. L'automatisation de la circulation des documents assure la coordination entre les tâches.

On distingue deux types d'approches dans les applications workflow. L'approche orientée procédure est basée sur des règles de routage des documents au sein d'une organisation. Elle est illustrée sur le marché par le système Link Works]. L'approche orientée formulaire est basée sur l'identification de types de champs à partir de la structure des documents et la définition de rôles fonctionnels associés aux individus susceptibles de remplir ces champs. Des règles définissent les conditions sous lesquelles les champs peuvent être saisis et consultés. Cette approche est illustrée sur le marché par le système Lotus Notes.

5.4 Systèmes intégrés

Les systèmes intégrés offrent aux individus des environnements complets pour communiquer, produire et se coordonner. Ils se divisent en deux familles: les salles de réunion aménagées et les espaces de travail partagés. Les salles de réunion aménagées utilisent la présence dans une même salle d'équipements d'aide au travail collaboratif et d'individus pour intégrer les activités collaboratives correspondant aux facettes du trèfle. Les espaces de travail partagé fournissent l'accès

à plusieurs applications juxtaposées dans la même fenêtre et/ou accessibles sur le même serveur. Le tableau suivant présente des exemples type d'applications partagées à l'aide des systèmes intégrés.

Application	Description
Liste d'idées	Application pour éditer des listes de messages représentant des idées, avec éventuellement moyen de classer ces idées par catégories, par mots-clés, etc.
Graphe d'idées	Comme une liste d'idées, mais présentée sous forme de graphe. La structure du graphe sert à associer les idées. Elle impose une forme de coordination entre les actions des utilisateurs pour qu'ils ajoutent les nouvelles idées aux bons endroits.
Tableau partagé	Editeur partagé de dessins et de textes disposés librement sur une surface infinie. Les tableaux partagés sont souvent non structurés, pour servir au "brains-torming". Parfois le tableau est capable d'interpréter et de traiter les données saisies à l'aide d'algorithmes de reconnaissance de gestes. Le tableau peut traduire directement les données dans un formalisme adapté au type de la réunion (par exemple en réseau de Pétri).
Outil de vote	Application pour soumettre des questions et des propositions à un vote ou à une évaluation puis traiter les résultats.
Partage d'application mono-utilisateur	Application pour répliquer une application mono-utilisateur sur plusieurs sites. L'interaction s'effectue par la prise de contrôle de l'interface à distance qui n'est possible que par un seul utilisateur à la fois. Le partage nécessite un mécanisme de coordination pour prendre le contrôle de l'application. Certains systèmes permettent également de superposer des annotations sur les fenêtres de l'application répliquée. Le partage d'applications non conçues initialement pour être partagées est un sujet de recherche actif. Une approche possible consiste à utiliser un gestionnaire de fenêtre qui intercepte et distribue sur plusieurs sites les événements destinés à l'application.
Importation	Application pour afficher une image, une présentation Power Point, ou pour donner un fichier, etc.
Vidéoconférence	Transmission de l'image et du son.
Dialogue textuel et messagerie	Outils d'échange de messages semblables aux outils de CMC.

Tableau 5 : exemples type d'applications partagées à l'aide des systèmes intégrés

Salles de réunion aménagées

Les groupes face-à-face bénéficient des collecticiels sous la forme de salles de réunion aménagées, ou bien de salles spécialisées en fonction de la tâche du groupe. Le regroupement physique des personnes dans une même salle permet de communiquer, de produire et de se coordonner avec les canaux traditionnels, comme la voix ou les gestes tout en tirant partie d'équipements informatiques. Deux types d'équipements sont employés: d'une part un ou plusieurs écrans de grande taille visibles par tous les participants, d'autre part des ordinateurs personnels ou d'autres périphériques connectés sur le réseau de la salle.

Espaces de travail partagés

Les espaces de travail partagés intègrent les facettes du trèfle car ils contiennent des informations ou des applications appartenant aux trois espaces du trèfle. Ils existent soit de manière permanente, soit de manière temporaire le temps d'une session de travail. Les informations contenues dans les espaces de travail partagés sont accessibles avec des logiciels clients spécialisés lorsqu'ils sont hébergés sur un serveur ou sinon par connexion directe entre les logiciels des participants. Le contenu de l'espace partagé est modifiable comme avec les éditeurs partagés. Par rapport à un système de gestion de fichiers multi-utilisateurs les espaces partagés transmettent plusieurs informations supplémentaires parmi lesquelles:

- la présence des autres utilisateurs;
- les activités en cours de chacun;
- l'historique des activités passées;
- l'identité des utilisateurs;
- des communications entre utilisateurs.

Les espaces partagés sont distingués en trois classes en fonction de la métaphore d'accès à leur contenu :

- Première classe : Document,
- Deuxième classe : pièce,
- Troisième classe : réunion.

La première classe est basée sur une extension des serveurs web. Elle utilise le protocole de transmission HTTP et complète les documents HTML par des informations de conscience mutuelle (les quatre premiers types d'information). La seconde classe est basée sur des serveurs utilisant des protocoles de communication spécialisés. Ils regroupent l'accès à des applications de réunion à travers une métaphore de pièces. Enfin la troisième classe correspond aux systèmes de conférence électronique dont l'accès est basé sur l'organisation de réunions. Comme exemple nous pouvons citer le système BSCW de la première classe, TeamWave Workplace de la deuxième classe et PictureTel de la troisième classe.

6 Outils pour la mise en œuvre d'un collecticiel

Dans cette section nous présentons une catégorie d'outils dédiés à la conception et de réalisation logicielles des collecticiels. Nous présentons 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 collecticiel. 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 collecticiel, l'espace IHM, et la conception et réalisation logicielles.

6.1 Cycle de vie logiciel

Les étapes du cycle de vie en V sont représentées à la Figure 14. Il existe d'autres cycles de vie comme le cycle en cascade ou en spirale. 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.

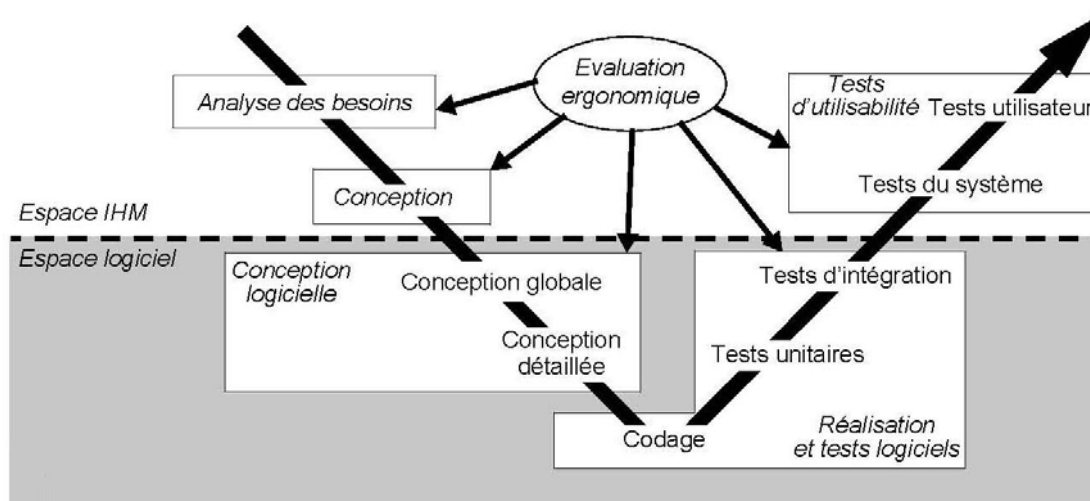


Figure 14 : Cycle de vie en V de développement logiciel (Laurillau, 2002)

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 9, 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.

- Analyse des besoins : 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. 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.
- 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.
- Conception logicielle : L'étape de conception logicielle est une étape préliminaire à la phase de codage. A l'aide d'outils pour la conception logicielle, cette étape consiste à construire l'architecture logicielle décrivant la structure du logiciel à développer ainsi que son comportement à l'exécution.
- Codage : La finalité de cette étape est de produire un logiciel exécutable.

6.2 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. Nous présentons cette étape dans le cas des collecticiels 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.

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. 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 (prendre 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. L’étude ethnographique est une forme de reportage bien plus qu’une collecte de données et l’ethnographe interprète ce qu’il observe (Laurillau, 2002). 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 collecticiel 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, à priori, aucune limitation de ce nombre.

Il existe d’autres approches qui tendent à intégrer les apports de l’ethnographie dans la réalisation de collecticiels à travers la notion d’action située (une action doit être considérée par rapport à son contexte d’exécution), ou comme les travaux issus de l’école scandinave à travers la notion de conception participative (participatory design). Cette dernière approche vise à impliquer les utilisateurs tout au long du cycle de développement d’un collecticiel.

Le Modèle de Denver

Le modèle de Denver offre un cadre d’évaluation et de conception des fonctionnalités pour les collecticiels. Il résulte de la mise en commun de travaux de quatorze participants à l’atelier sur « la conception et l’évaluation des collecticiels », tenu pendant la conférence *ACM CHI (Computer Human-Interaction)* en 1995 à Denver (Colorado). Ce modèle comporte trois sous-modèles qui forment trois couches superposées : en haut, un modèle des objectifs et des besoins (*System*

Requirements), au milieu un modèle de conception (*Design Model*) et en dessous un modèle technologique (*Technology*). Le développement principal du modèle s'est porté sur le modèle intermédiaire, le modèle de conception. Le modèle de conception définit un cadre de conception d'un collecticiel, par cinq catégories de caractéristiques :

- Les Personnes (*People*) : cette catégorie inclut les caractéristiques des utilisateurs, des groupes et des rôles qu'une personne peut avoir au sein d'un groupe. Une personne peut être identifiée par son nom, son adresse, son numéro de téléphone, son métier, etc. Son rôle peut être identifié par rapport à un groupe, par rapport à la gestion du collecticiel, ou d'autres facteurs plus subjectifs. Chaque personne peut appartenir à plusieurs groupes et tenir un rôle différent dans chacun d'eux. Les caractéristiques du groupe vont de « stable et homogène » à « instable et hétérogène ». Les auteurs du modèle de Denver notent que la représentation du groupe reste une question ouverte.
- Les Artéfacts (*Artifacts*) : cette catégorie se réfère à tous les objets manipulés, produits ou consommés pendant l'interaction. Il en existe cinq types génériques : le texte, le son, l'image temporel, l'image statique et l'élément de calcul. Ces types peuvent se combiner à l'intérieur d'un artefact qui est alors composite. Les artefacts possèdent des attributs comme la coterporalité, la révisabilité, la possibilité d'en faire un compte-rendu, etc. Par exemple les différents modèles conceptuels, l'inventaire des besoins en données sur le système réel, les différents modèles opérationnels, les différents cadres d'expérimentations et la documentation du projet de simulation constitueront les résultats produits par les équipes du projet.
- Les Tâches et Activités (*Tasks and Activities*) : cette catégorie regroupe les caractéristiques des tâches et activités réalisables par le biais du collecticiel. Le modèle de Denver distingue quatre niveaux : les objectifs, les tâches et scénarios, les activités et les opérations. Les objectifs correspondent aux objectifs les plus globaux. Dans le domaine de la simulation, les objectifs sont par exemple «évaluation de performances d'un système», « respecter les délais et les coûts tout en garantissant la qualité d'étude souhaité », etc. À un niveau inférieur, les tâches sont par exemple «concevoir les modèles», « valider des documents », etc. Les activités se combinent pour former des tâches : « actualiser sa base de modèles », « programmer des modèles », « échanger des idées » en sont des exemples. Enfin, les opérations sont des manipulations élémentaires sur des artefacts.

- Les Situations d'Interaction (*Interactive Situations*) : Elles définissent la relation des participants entre eux, leur relation au temps et leur relation à l'espace. Les auteurs du modèle les représentent par une étoile à cinq axes qui correspondent respectivement à cinq notions : la dépendance, la temporalité, la coordination temporelle, le localisation du groupe et la taille du groupe. Ainsi, les caractéristiques du collecticiel permettent de donner la position de cinq points appartenant à chacun de ces axes (le degré d'entropie maximum est situé au centre de l'étoile). Les cinq points obtenus, reliés entre eux, forment une figure à cinq côtés qui est une des « signatures » du collecticiel.

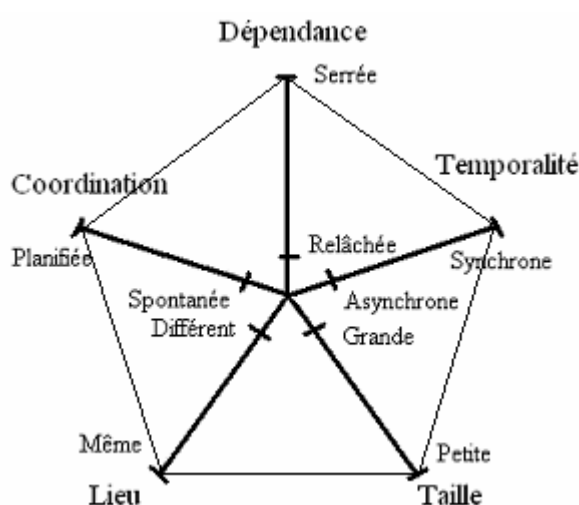


Figure 15 : Les cinq axes caractérisant la situation d'interaction (Salvador et al., 1996)

- Les Protocoles Sociaux de l'Interaction (*Interactive Social Protocols*) : un protocole social se réfère aux séquences possibles d'échange de signaux et d'informations qui déterminent et identifient les conflits et ses résolutions. Les auteurs du modèle utilisent la même représentation en étoile à cinq axes que pour les situations interactives. Ici les axes correspondent au style de réunions (meeting style), à la détection de conflits et à leur résolution (contention detection and resolution), à la taille du groupe, à la réglementation des interventions (floor control) et à la familiarité (formality of address). Le style de réunion décrit la façon dont les réunions entre participants se déroulent (unidirectionnel ou multidirectionnel). La caractéristique intitulée « détection des conflits et résolution » décrit la capacité du collecticiel à découvrir les conflits éventuels au sens large, le conflit est une situation de concurrence ou d'opposition des participants et le moment de leur résolution. La réglementation des interventions se réfère à la façon plus ou moins dirigée d'agir ou par exemple de « prendre la parole ». La familiarité correspond quant à elle à la façon plus ou moins formelle et protocolaire qu'ont les participants de s'adresser les uns aux autres.

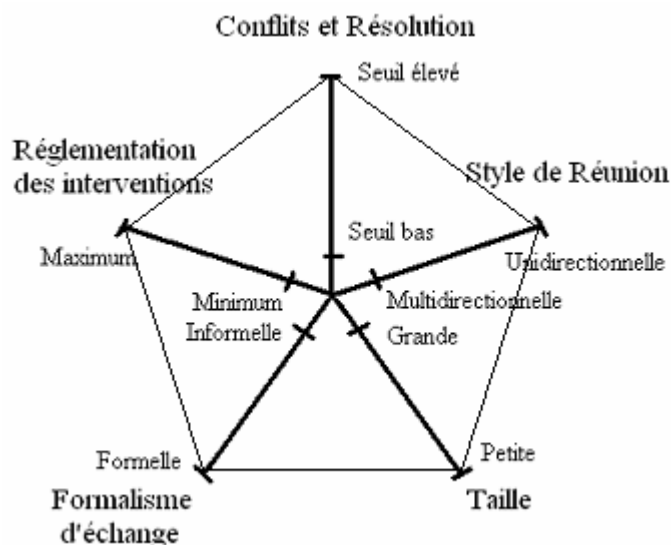


Figure 16 : Les six axes caractérisant le protocole d'interaction (Salvador et al., 1996).

La méthode GTA et l'analyse de la tâche

La méthode GTA (Groupware Task Analysis) est une méthode pour l'analyse de tâche du groupware. A l'issue de cette étape, une structure hiérarchique des tâches est souvent obtenue. 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, par exemple, des formalismes orientés objets comme MAD (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 passons à l'étape de conception et de spécification du système

6.3 Outils pour les spécifications fonctionnelles et externes

A partir de l'analyse de besoins, il convient alors de concevoir le modèle de l'interaction. Les spécifications externes d'un collecticiel s'expriment en termes de requis fonctionnels et requis utilisateurs. Les requis fonctionnels cernent les fonctionnalités offertes par le collecticiel. Le modèle du trèfle d'un collecticiel propose un cadre pour organiser et définir les fonctionnalités d'un collecticiel. Les requis utilisateurs concernent l'utilisabilité du collecticiel et donc l'interface du collecticiel. Communément, l'utilisabilité se décline en propriétés ergonomiques.

Le modèle du trèfle fournit un cadre conceptuel utile pour déterminer les requis fonctionnels et mener une analyse fonctionnelle. Les trois espaces fonctionnels d'un collecticiel étant définis, il convient néanmoins de noter que certaines fonctionnalités peuvent être à l'intersection de plusieurs 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. De plus, le rôle fonctionnel global d'un collecticiel 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 collecticiel selon la production, la communication et la coordination. Par exemple, Franck TARPIN (Tarpin-Bernard, 1997b) propose une méthode pour spécifier les fonctionnalités d'un collecticiel en fonction d'un ensemble de questions thématiques, en s'appuyant sur les trois espaces fonctionnels du modèle du trèfle :

- *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).

6.4 Spécifications externes et propriétés ergonomiques

Les 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 collecticiel doit répondre à des exigences et à un niveau de qualité requis. En génie logiciel, il existe de nombreux facteurs pour quantifier la qualité 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. Ce facteur définit la qualité d'utilisation d'un système interactif et sert de guide pour la conception et l'évaluation. Une approche en IHM 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). Ces 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. 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 :

- **Conscience de groupe et rétroaction de groupe** : lors d'une activité en groupe les membres ont toujours un œil 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 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. 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** : la propriété de protection de la vie privée est relative à la protection des informations privées. 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. 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. Certains auteurs désignent cette propriété par la notion d'observabilité publiée. Des informations observables de l'utilisateur, peuvent être sous trois formes d'observabilité : Observable (ce qui est souhaitable qu'un utilisateur perçoive), Publiable (ce qu'un utilisateur décide de rendre observable de lui-même et Filtrable (la forme de ce qui est observable).

- **Réciprocité 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 et couplage de l'interaction** : La propriété de *WYWISIS* a pour but d'assurer la rétroaction du groupe, c'est-à-dire de permettre à chaque participant de voir ce qu'un autre participant voit ou fait, se traduit en principe 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. 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 tandis que le mode *WYSIWIS* relâché aura tendance à favoriser la protection de l'espace privé.
- **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, c'est-

à-dire 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.

6.5 Outils pour la conception logicielle

Le développement du logiciel se déroule en deux grandes étapes : 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. 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. 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, PAC* ou le modèle de Dewan. Ces modèles définissent des structures modulaires normalisées. 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.

6.6 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 oeuvre de ces systèmes (accès concurrents, perte d'information sur le réseau, etc). L'ensemble des outils déjà disponibles pour la construction d'un collecticiel est très large. Ces outils sont classés en huit catégories :

1. **Les bases de données relationnelles**, qui proposent des outils d'accès concurrents à des données partagées.
2. **Les systèmes distribués**, qui permettent d'utiliser des mécanismes de haut niveau d'accès à des informations réparties (ex : les RPC, CORBA, DIS, HLA).
3. **Les serveurs de messages**, qui relayent des informations et peuvent fonctionner de façon anonyme (ex : CB Message Bus).
4. **Les systèmes à fenêtres partagées**, qui sont l'extension primaire des environnements mono-utilisateur (ex : SharedX).
5. **Les systèmes à objets partagés**, qui permettent les échanges de données entre les composants partagés des objets (ex : Colab). Ces outils, comme ceux qui suivent, reposent généralement sur les trois premiers types d'outils.
6. **Les boîtes à outils multi-utilisateur**, qui assurent certaines fonctionnalités de collaboration (ex : DCS pour des systèmes de conférence) ou proposent des fonctions de bas niveau (ex : MMConf ou GroupKit).
7. **Les UIMS multi-utilisateurs**, qui étendent les propriétés des User Interface Management Systems mono-utilisateur (ex : Rendezvous).
8. **Les générateurs d'interfaces multi-utilisateurs**, qui permettent de générer des interfaces partagées (ex : Suite).

Tous ces systèmes permettent de mettre en place des applications avec lesquelles plusieurs utilisateurs peuvent travailler simultanément avec plus ou moins de conscience de la présence des autres. Ils apportent des éléments de réponse au vaste problème que constitue le TCAO, et sont adaptés à certains modes de coopération. De nombreux auteurs ont déjà comparé les fonctionnalités et performances de ces outils.

6.7 Outils pour l'évaluation ergonomique

L'évaluation ergonomique intervient à toutes les étapes du cycle de vie logicielle, quel que soit le type de système interactif (mono utilisateur ou collecticiel). L'évaluation ergonomique permet de détecter les problèmes d'utilisabilité du collecticiel à priori lors des phases de conception ou à 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 et les techniques expérimentales. Le premier type de techniques repose sur des modèles théoriques tels que le modèle *ICS* (*Interactive Cognitive Subsystems*, modèle cognitif de l'utilisateur) ou des modèles de performance comme, par exemple, le modèle *GOMS* (*Goals, Operators, Methods and Selection Rules*), ainsi que l'utilisation d'heuristiques. Ces heuristiques peuvent être liées à des propriétés ergonomiques. 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 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 visant à adapter aux collecticiels la technique d'évaluation heuristiques. Certain auteur propose un ensemble de huit heuristiques issues de la théorie de la mécanique de la collaboration. 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 :

- Un collecticiel doit fournir des moyens pour rendre observable l'échange verbal d'informations.
- Un collecticiel doit fournir des moyens pour rendre observable l'échange par la gestuelle d'informations.
- 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.).
- Un collecticiel doit fournir des moyens pour rendre observable l'échange d'informations induit par la manipulation d'artéfacts partagés (manipulation d'objets partagés et retour d'information sur les changements d'états de ces objets).
- Un collecticiel doit fournir des moyens pour protéger l'espace de travail, l'espace privé, les objets partagés, etc.
- Un collecticiel doit permettre plusieurs modes d'interaction (fortement couplé ou faiblement couplé) et la possibilité de modifier dynamiquement le niveau de couplage.
- Un collecticiel doit mettre en oeuvre des moyens pour autoriser les participants à coordonner leurs actions.
- 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. Cette approche a l'avantage d'offrir une méthode d'évaluation à faible coût qui est 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.

7 Réalisations existantes

La recherche en TCAO a débuté vers la fin des années 1980, depuis beaucoup de concepts et de réalisations ont été élaborés. Dans cette section nous allons présenter un échantillon des différents projets conçus autour de la collaboration et qui répondent à un, ou plusieurs, des critères suivants :

- capable de supporter le travail coopératif dans une équipe de projet de simulation qui est basé essentiellement sur la production conjointe des documents et le partage d'application.
 - importance dans la recherche en TCAO.
 - en relation avec la collaboration sur le Web.
- **BSCW Shared Workspace** (Payet,2003) : Basic Support for Cooperative Work (BSCW), est un projet du GMD qui a pour objectif de définir des outils collaboratifs sur le Web qui fournissent des services collaboratifs inter-plates-formes à des groupes exploitant des technologies Web existantes. Ce système tente à fournir des caractéristiques basiques pour la coopération dans des services intégrés, accessibles depuis différentes plates-formes et qui ne demande pas à l'utilisateur d'adopter des applications spécifiques. Le BSCW Shared Workspace est un outil du BSCW, c'est une application coopérative centralisée intégrée à un serveur Web non modifié et accessible depuis des navigateurs Web standard. Ce système supporte la coopération à travers des 'espaces partagés', formant des petits référentiels dans lesquels l'utilisateur peut télécharger des documents, suivre le fil d'une discussion, et obtenir des informations sur les activités précédentes des autres utilisateurs pour pouvoir coordonner son propre travail. Dans ce type de système c'est l'utilisateur qui doit prendre l'initiative de demander les informations nécessaires à la coordination de ses propres activités. Le système permet le partage de document et quelques formes de communication, mais n'est pas en mesure d'accueillir des fonctionnalités collaboratives plus variées.
- **TANGO** (Beca et al., 1997) : Les technologies WEB actuelles offrent des opportunités remarquables aux nouveaux environnements sophistiqués de collaboration via le réseau Internet ou Intranet. La technologie de base est un ensemble distribué de serveurs et de clients Java qui communiquent entre eux. Les clients Java sont des applets lancés depuis un navigateur ordinaire. Ces environnements peuvent être réalisés comme un ensemble de processus de communication avec les serveurs coordonnant le trafic entre les clients. Dans un environnement de collaboration traditionnel, les serveurs lancent les contrôleurs de session qui contrôlent l'accès des nouveaux utilisateurs, et assurent une vue commune de l'environnement de sorte que les applications engendrées sur un client (tableaux blancs, leçons pour le télé-enseignement, etc. ...) soient partagées par tous les autres utilisateurs. Ce partage simple d'information est déjà un environnement puissant. La généralisation d'un tel environnement par l'ajout des filtres pour traiter les messages des clients avant qu'ils ne soient reçus par les destinataires permet aux différents utilisateurs de recevoir différentes vues des mêmes données. Ces filtres peuvent être gérés par des scripts dynamiques qui

permettent d'inclure les utilisateurs de la simulation pour des applications de formation et de modélisation. Notons que les utilisateurs produisant et recevant des messages peuvent être des utilisateurs ou des ordinateurs. Parmi les travaux s'inscrivant dans cette perspective l'environnement TANGO. Il est généralisable, flexible et extensible. Il permet la création de diverses zones d'information partagées et offre un support de haute qualité pour la visualisation et l'interaction avec l'utilisateur basé sur le travail collaboratif. De plus, il offre un support pour des vues indépendantes et coordonnées des différents utilisateurs connectés. TANGO vise à étendre le paradigme du WEB au domaine du TCAO et bien au-delà du concept de chatting, de tableau blanc partagé et de partage d'applications génériques simples.

- **WorkPalce (Teamwave)** (Roseman et Greenberg, 1997) : est un environnement collecticiel qui supporte une large variété d'activités collaboratives. Il intègre dans un seul environnement des tableaux blancs partagés, des possibilités de la messagerie instantané, et des composants collecticiel personnalisés comme les notes, les bases de données et les calendriers. Le système offre un environnement de travail persistant qui supporte en même temps le travail collaboratif synchrone et asynchrone. Le WorkPlace est structuré autour d'une métaphore basée sur la notion des salles. Comme dans des salles réelles, une équipe de travail ou une communauté des utilisateurs maintient une ou plusieurs salles qui servent de point focal à leurs collaborations. Si plusieurs personnes occupent une salle en même temps, elles peuvent s'engager dans une collaboration synchrone. Si une salle est occupée par une seule personne, elle peut travailler seule, et laisser probablement l'information pour d'autres quand elles entrent dans la salle à leur tour. De cette façon, elles supportent le travail asynchrone. Les utilisateurs peuvent apporter arbitrairement des outils à la salle en la personnalisant pour convenir à leurs besoins. Quand les personnes quittent une salle, son contenu persiste et reste en place pour le prochain utilisateur qui entre dans la salle. Les salles électroniques dans le Workplace supportent des utilisateurs qui ne sont pas physiquement présents dans la salle. Les salles d'une équipe de travail sont hébergées sur un serveur. Les membres du groupe utilisent le client du TeamWave pour se connecter au serveur à travers Internet. C'est là où ils peuvent interagir avec les salles et avec d'autres utilisateurs présents. Les salles contiennent des outils standard tels que des tableaux blancs partagés et de la messagerie instantanée, et autorisent l'addition des outils personnalisés pour des besoins particuliers du groupe (ex: Outils de simulation); chacun de ces derniers constitue une application entièrement interactive et faisant partie du collecticiel. Les clients et les serveurs sont déployés sur Macintosh, Windows et plusieurs plateformes Unix. La

communication se fait avec le protocole TCP/IP, que se soit en utilisant une liaison Internet réseau local LAN/WAN ou par l'intermédiaire d'une connexion Internet via un modem classique. Un dépôt persistant garde l'état des salles et permet la récupération des anciennes versions des états de la salle.

- **CVW** (Peter et al., 1997) : Le CVW (Collaborative Virtual Workspace) ou espace virtuel de travail collaboratif. C'est un environnement conçu pour supporter un travail d'équipes dispersées géographiquement ou dans le temps. Du point de vue de l'utilisateur, CVW fournit un espace virtuel persistant dans lequel les applications, les documents et les utilisateurs existent dans des Salles, des étages et des bâtiments. D'un point de vue technique, c'est un cadre pour intégrer diverses possibilités de collaboration. Pour un utilisateur, CVW est un bâtiment divisé en plusieurs étages et salles, où chaque pièce fournit un contexte pour le partage de communication et de document. Des utilisateurs réunis dans des salles peuvent "chatter" par messagerie instantané, discuter par conversation audio, partager de la vidéo. En définissant des salles comme étant la base de la communication, signifie que des utilisateurs n'ont pas besoin d'établir des sessions ou de connaître la localisation de l'autre utilisateur; ils ont besoin d'entrer dans une salle tout simplement. Si les utilisateurs choisissent de communiquer par audio, vidéo ou texte, alors la session de communication est établie automatiquement pour eux. Les utilisateurs peuvent verrouiller les salles et communiquer en privé à l'intérieur d'une salle ou entre les salles. Les salles servent également de base au partage de document. Les utilisateurs peuvent apporter des documents avec eux, et peuvent laisser différents types de documents dans une salle, permettant à n'importe qui dans cette pièce de lire le document. La persistance est supportée car les salles existent même lorsque personne ne s'y trouve. En conséquence, le document demeure dans la salle pour une consultation par des futurs visiteurs jusqu'à ce qu'un utilisateur autorisé l'efface ou le déplace. Les types de document incluent des tableaux blancs, des URLs, des notes et d'autres documents édités par les applications locales de l'utilisateur (par exemple, traitement de textes, tableurs). Les documents qui peuvent être édités par des applications locales sont contrôlés par un système de document dans le CVW.

8 Conclusion

Dans ce chapitre que nous avons présenté les collecticiels, ainsi que leur domaine. Nous avons étudié leurs objectifs, leurs caractéristiques et les réalisations qui ont été faites jusqu'à présent. Nous avons également examiné les outils existants de conception et de développement de ce type d'application.

En se basant sur les résultats du deuxième et du troisième chapitre, le quatrième chapitre vise à concevoir un collecticiel de simulation. Cet outil sera ensuite implémenter pour le mettre en situation d'expérimentation dans le chapitre 5.

Chapitre n° 4

*Modèles de
Collecticiel de
Simulation*

IV Modèles conceptuels d'un collecticiel de simulation

1 Introduction

Après avoir défini les principes du collecticiel puis leurs outils de conception et de développement logiciel dans le chapitre précédent, nous allons maintenant appliquer cette connaissance dans le domaine de la simulation en vue de proposer sur la base des résultats du deuxième chapitre, un modèle de collecticiel de simulation et mieux cerner les attentes des utilisateurs du domaine. Cette mise en application vise à concevoir le modèle d'architecture conceptuelle et logiciel du collecticiel visé. Nous validons ensuite cette conception par une étude expérimentale.

Avant de tenter une transposition des théories avancées dans le domaine du TCAO, il convient de revenir sur les rapports entretenus entre sciences sociales et informatisation des relations entre utilisateurs d'un système coopératif ou collaboratif. L'intérêt porté par la communauté informatique aux sciences sociales semble lié aux critiques croissantes faites à l'encontre du mode de conception utilisé dans les années 1980. Les préoccupations des concepteurs se tournent alors vers l'utilisateur et les rapports que celui-ci entretient avec son contexte d'action afin d'entreprendre une réflexion de fond sur la nature de l'assistance informatisée au travail coopératif. Parmi les différents courants émergeant depuis deux décennies dans le domaine du TCAO, il semble possible d'isoler quatre types de démarches (Hanser, 2003) :

- Basées sur la théorie de l'activité, elle-même héritée de la psychologie comportementale russe. Elle est principalement mise en œuvre afin de décrire les interactions entre acteurs coopérant ou collaborant au cours d'une activité commune.
- Basées sur la théorie de la structuration de Giddens, et proposant une analyse fondée sur les rapports hiérarchiques entre individus.
- Basées sur la théorie des réseaux d'acteurs, analysant les chaînes de représentation et l'historique des échanges ayant lieu entre acteurs d'un système.
- Basées sur la théorie de l'interaction (ou action située), détaillant les mécanismes d'interaction et d'articulation des activités individuelles au cours d'un travail collaboratif.

Ce chapitre sera pour nous l'occasion d'appliquer ces méthodes, en nous focalisant toutefois sur les méthodes de la théorie de l'activité et la théorie de l'action située qui trouvent un écho grandissant dans la communauté informatique.

La première partie de ce chapitre est consacrée à l'analyse des besoins et à l'étude conceptuelle du collecticiel. Cette analyse comprend une caractérisation du collecticiel avec le Denver model, l'élaboration du modèle de tâche avec la méthode GTA, l'analyse des besoins fonctionnels et l'analyse des besoins ergonomiques. Dans la deuxième partie, nous présentons notre approche de l'élaboration du modèle d'architecture logicielle. Dans la dernière partie, nous présentons le modèle et les spécifications externes de l'Interface utilisateur.

2 Hypothèses

Notre connaissance de la nature des projets de simulation nous amène à formuler une hypothèse fondatrice à notre proposition. Notre première hypothèse est qu'il faut que le modèle coopératif du collecticiel futur doive correctement supporter le contexte propre à l'entreprise-projet, que nous jugeons caractéristique d'un projet de simulation. La technologie issue des collecticiels projet utilisant internet nous paraît en adéquation avec le concept de l'entreprise-projet ; l'universalité du réseau internet et le fait qu'il n'appartienne à aucune organisation sont en rapport direct avec les qualités de l'entreprise-projet, ouverte et éphémère.

Nous allons maintenant exposer deux sous-hypothèses corollaires de cette première, plus précises quant au fonctionnement même du collecticiel que nous allons proposer.

La première concerne l'identification des mécanismes moteurs et structurants de la coopération dans ce contexte. Plus trivialement posée, la question pourrait se formuler ainsi : comment un acteur connaît-il, à tout moment, la part de ce qu'il lui revient à faire dans l'action collective ? Nous voyons deux réponses possibles à cette question :

- L'acteur sait ce qu'il doit faire parce qu'on le lui a dit. Nous parlons donc de coordination explicite. Étant donné que le travail individuel d'un acteur se compose principalement de phases solitaires dans lesquelles l'acteur est isolé de ses partenaires pour produire une part de l'information globale, il y a nécessité pour lui et pour le groupe auquel il appartient d'anticiper les éventuelles divergences que ce mode de fonctionnement risque d'entraîner. Il faut que le futur collecticiel offre les mécanismes de coordination explicite, programmée et spontanée pour ces utilisateurs.
- L'acteur sait par expérience ce qu'il doit faire. Nous parlons ici de coordination implicite ou auto-coordination. L'expérience de chaque acteur l'aide à repérer et à délimiter son champ d'intervention sans que personne n'ait à lui rappeler ou à lui

indiquer expressément. Un chef de projet, par exemple, pour anticiper une difficulté technique qu'il a déjà rencontré, peut mettre à disposition à un participant qui doit la résoudre, l'information la plus précise et la plus pertinente. Par définition, la coordination implicite ne peut être que spontanée (et non programmée).

Nous pensons que ces réponses ne sont pas contradictoires, et que le collecticiel que nous projetons doit reproduire cette double réalité. Il reste à déterminer l'articulation et la place relative des mécanismes de coordination explicite et de coordination implicite dans le collecticiel. Si l'on considère la pratique de la conduite d'un projet de simulation, ces deux mécanismes prennent place à des niveaux différents de gestion du projet : la coordination explicite semble particulièrement vouée à donner des moyens de contrôle formels sur les grandes lignes d'une opération (ex. validations de phases), alors que la coordination implicite prend place face à des situations de problème courantes et fortuites (ex. coup de téléphone pour régler rapidement une difficulté).

La coordination implicite, s'appuyant sur une conception optimiste de la relations entre acteurs, et à condition de leur donner toute l'information contextuelle nécessaire pour juger des situations face auxquelles ils sont confrontés, peut être le ressort principal de l'activité collaborative. La coordination explicite, par la mise en œuvre ponctuelle de mécanismes déterminés (réunion, planning, ordre formel) vient la suppléer pour garantir, la bonne marche des opérations lorsque la confiance et la responsabilisation des acteurs ne suffit plus. De façon imagée, la coordination implicite constitue le socle de notre modèle coopératif tandis que la coordination explicite prend appui sur elle. Cette prééminence de la coordination implicite dans notre proposition peut être considérée aussi comme une suite logique du choix que nous avons fait d'un collecticiel support de l'entreprise-projet.

La deuxième hypothèse est que le futur collecticiel doit permettre à chaque utilisateur de puisse travailler sur sa machine de façon isolée et sans se soucier de règles communes, et néanmoins de lui permettre de partager des données avec des partenaires, tout en lui garantissant un niveau élevé de confidentialité, d'intégrité des données et de conscience de groupe.

3 Exigence pour la conception de collecticiel

La nature du travail collaboratif impose un certain nombre d'exigences tant aussi bien au niveau des caractéristiques de l'application support, qu'au niveau du fonctionnement et des procédés qui permettent la conception de ce type d'application. Dans la suite nous allons présenter ces différentes exigences.

3.1 La malléabilité

L'imprévisibilité des besoins qui vont émerger au cours du travail collaboratif, fait qu'il est nécessaire que les systèmes logiciels doivent supporter un travail coopératif sensible au contexte dans lequel ils seront utilisés. Ce contexte s'exprime, par exemple, par la situation coopérative qui se présente (organisation, procédures, style de coopération), par la spécificité de la tâche à réaliser, et par les paramètres dans lesquels la coopération prend place (distribution géographique, base logicielle et matérielle, connections réseaux). Cela engendre une certaine "versatilité" des exigences, et le système collaboratif doit être en mesure de supporter cela.

Le terme malléabilité est la traduction du terme "tailorability" : ce terme exprime la flexibilité et l'adaptabilité d'une application face aux nouveaux besoins auxquels elle est soumise.. Ainsi une application possédant la propriété de "malléabilité" sera appelée une application "malléable".

Les deux principales caractéristiques de la malléabilité sont les suivantes :

- La malléabilité ne doit pas être accessible qu'aux seuls développeurs du logiciel, mais doit pouvoir surtout être réalisée par l'utilisateur final ou par un responsable plus expérimentée (administrateur).
- Sa mise en œuvre doit faire partie de l'utilisation normale du système, et non être une activité de développement séparée de l'utilisation de l'application.

Aussi, on peut distinguer deux formes de malléabilité :

- malléabilité programmeur : réalisée par un programmeur expérimenté, elle peut conduire aux développements de nouveaux outils à intégrer au système.
- malléabilité pour l'utilisateur final : qui traduit les possibilités qu'ont les utilisateurs finaux pour étendre et adapter le système de sorte à l'accorder à leur besoin.

3.2 Multi-personnalisation

Pour pouvoir supporter l'intégration dynamique du travail coopératif formel et informel, un système TCAO doit posséder un mécanisme d'incorporation accessible aux utilisateurs, et doit aider les utilisateurs à interpréter ce mécanisme et à évaluer sa justification et ses implications. Il doit assister l'utilisateur dans l'application et l'adaptation de ce mécanisme à la situation en cours. En plus

de cela, un outil destiné à un groupe d'utilisateurs doit pouvoir être accepté par tous les membres du groupe. Or chaque individu a sa propre sensibilité, sa propre perception des choses. Le système doit être suffisamment adaptable pour pouvoir supporter de nouvelles façons d'être utilisées, façons non prévues par le développeur.

Les exigences des utilisateurs ne se limitent pas seulement à des préférences d'apparence de l'interface graphique de l'outil, mais également des préférences au niveau des sous fonctions de l'application. Ainsi, il existe deux catégories de personnalisation : personnalisation d'interface et la personnalisation fonctionnelle.

La personnalisation d'interface concerne l'aspect visuel de l'application et les procédés d'interaction qu'elle supporte. L'utilisateur peut avoir des préférences visuelles, liées à ses propres goûts ou imposées par l'environnement logiciel auquel il est accoutumé. De nos jours la majorité des outils intègre des options de personnalisation qui répondent tout à fait à ce type d'exigences, on peut citer Office, etc.

La personnalisation fonctionnelle répond à des exigences qui concernent plutôt les besoins de l'utilisateur en termes de service rendu par l'outil. Généralement, ces exigences sont analysées au cours de la phase de conception du logiciel, et satisfaites au moment de son installation sur le poste du client, quand l'utilisateur sélectionne les options d'installation qui lui conviennent. Dans ce cadre, seuls les besoins initiaux sont concernés, et rien ne permet d'assouvir les exigences qui se présentent tout au long de l'exploitation du produit. Ces besoins émergents sont très courants dans le travail collaboratif.

3.3 L'ouverture

L'un des mécanismes de malléabilité les plus importants est l'extension d'une application en cours d'exécution pour changer son comportement. Et à de renchérir que les systèmes collaboratifs doivent être adaptables et extensibles en même temps que l'environnement de travail subit des variations de contexte. Le système doit supporter la réutilisation et la combinaison d'outils pour différentes catégories d'équipes. Au moment où leurs besoins changent, les membres de l'équipe doivent être en mesure d'étendre et de modeler leur environnement de travail de sorte à ce qu'il soit adapté à leur besoin. Le système doit être apte à intégrer des modifications à tout moment. Il doit être "ouvert".

Les applications collaboratives requièrent deux formes d'ouverture de la part de la base du système support : celui-ci doit être ouvert pour accueillir de nouvelles formes de structures de données et de contenus, et ouvert pour accueillir de nouveaux outils collaboratifs (Payet, 2003). Une liste de 27 exigences auxquelles doit répondre un environnement collaboratif présenté dans (Haake et al., 1999) , dont voici un bref assortiment :

- les nouveaux services introduits dans l'environnement doivent être orthogonaux aux autres services en cours d'utilisation (ne pas les altérer).
- supporter un ensemble ouvert de documents et d'outils.
- supporter un ensemble ouvert de possibilités d'organisation de l'espace de travail.
- supporter les transitions entre les organisations différentes de l'espace de travail.
- être capable de recevoir un ensemble ouvert de politique de coordination.
- offrir des supports pour le partage de données entre plusieurs utilisateurs.
- offrir des supports pour le couplage des interfaces utilisateurs pour tout outil (espace d'affichage partagé).
- la conscience du groupe doit pouvoir être produite pour toute combinaison d'outils.
- supporter un ensemble ouvert de canaux et de politiques de communication.
- supporter les transitions entre les différents canaux et politiques de communication.
- offrir des supports pour différents modes de coopération.
- supporter la transition entre les modes de coopération.
- supporter différents modes de travail (individuel, asynchrone, synchrone, etc.)
- offrir des supports pour permettre de basculer entre les différents modes de travail.

3.4 L'articulation du travail

La répartition géographique du processus collaboratif associé au fort potentiel d'ouverture que doivent présenter les collecticiels engendrent une dimension supplémentaire : celle de l'articulation du travail. L'articulation du travail est l'ensemble des activités nécessaires à la gestion de la nature distribuée du travail (Payet, 2003). L'espace de l'articulation du travail doit subvenir aux exigences suivantes :

- il doit améliorer la mécanique et les capacités de traitement de l'information humaine pour permettre à l'ensemble coopératif d'accomplir des tâches qui sont infaisables par un seul individu.

- il doit permettre de combiner les activités spécialisées de plusieurs travailleurs, chacun exploitant des outils, des techniques ou des routines spécifiques.
- il doit faciliter l'utilisation de plusieurs stratégies et heuristiques⁴ pour résoudre un problème donné.
- il doit faciliter l'exploitation de plusieurs conceptions et perspectives pour un problème donné afin de correspondre au mieux aux différentes natures de l'environnement coopératif.

3.5 Intégration au Web

Une collaboration est une activité qui est réalisée par un groupe d'individus. Une application qui supporte ce mode de travail doit permettre à ses utilisateurs de communiquer entre eux et d'interagir sur des documents ou des objets communs à travers un environnement partagé accessible quelque soit la localisation géographique des individus. Cette exigence de premier ordre nous amène inévitablement à considérer le Web comme l'environnement de prédilection des futurs systèmes collaboratifs. Cette dernière exigence est en accord avec nos hypothèses précédemment présentés.

4 Les modes de coopération

Pour construire une application de groupe en générale, il est nécessaire d'identifier les modes de coopération selon lesquels les membres pourraient interagir avec l'application et communiquer ainsi entre eux. Globalement, une application de groupe, doit supporter plusieurs canaux de communication qui permettront d'assurer respectivement :

Une communication Homme-Machine avec les applications classiques utilisées localement par chaque membre (traitement de texte, Tableur, etc.). Parmi ces applications figure nécessairement l'outil de modélisation et de simulation.

Une communication Home-Machine qui met en jeu la coopération et dont le but essentiel est la manipulation coordonnée des objets partagés par les membres du groupe et ce dans un vrai mode multi-utilisateurs.

Une communication Homme-Homme qui offre aux membres du groupe plusieurs modes de conversation : messagerie, chat, vidéophone etc. ...

Dans (Tarpin-Bernard, 1997b) Tarpin, définit quatre modes de coopération pour l'activité de conception coopérante. Ces quatre modes de coopération sont généralisables aux autres activités coopératives du monde industriel (ex. conception d'un système de production). Ces modes correspondent à des granularités d'interaction de plus en plus fines. Partant de cette généralisation, nous identifions que la conduite d'un projet de simulation peut se faire selon différents modes de coopération suivants :

- La coopération asynchrone : Dans le contexte d'une organisation, la coopération asynchrone correspond au mode de travail autonome. Chacun travaille de son côté puis soumet des rapports, comptes-rendus, etc. aux interlocuteurs adéquats qui, plus tard, feront part de leurs analyses. Dans ce mode de coopération la rencontre est informelle, et chacun travaille indépendamment les uns des autres, mais il n'est pas exclu qu'un participant puisse utiliser un travail fait par un autre (modèle, code, etc.) Les différents participants interagissent dans le projet en échangeant les données et en travaillant quand ils le peuvent (à des instants généralement différents : sans co-temporalité).
- La coopération en session : La coopération en session correspond à l'acte de présence au sein de l'organisation et peut être rapprochée de la notion d'astreinte. En effet, dans un tel mode, les participants connaissent la liste de leurs interlocuteurs potentiels à un instant donné. Dans le cas de l'astreinte, cette liste est prédéfinie par le chef de projet. Elle se met à jour à partir des agendas individuels. A titre d'exemple, le téléphone peut représenter un outil type de la coopération en session. En effet, la communication téléphonique ne peut s'établir si les interlocuteurs ne sont pas présents simultanément. Cependant, l'informatique permet d'envisager des outils beaucoup plus puissants que le téléphone et qui permettent d'abolir d'une façon plus nette les barrières spatiales séparant les différents participants d'un projet. Dans ce mode de coopération les différents participants travaillent en même temps, mais de façon autonome. Ils sont accessibles pour communiquer (en co-temporalité), mais sans partager de façon visuelle les objets de leurs discussions.
- La coopération en réunion : La notion de réunion introduite dans la coopération en réunion correspond exactement à celle qui existe dans le cadre d'un projet. De fait, elle est planifiée et elle définit des rôles spécifiques à chaque participant. De plus, elle porte sur un ensemble d'objets clairement identifié (Modèle, Programme, Cadre d'expérimentation, document, etc.) qui

doivent être partagés par l'ensemble des membres de la réunion. Contrairement à la coopération asynchrone, la coopération en réunion impose la participation active (ex: réponse immédiate aux questions). Dans ce mode coopération des participants clairement identifiés travaillent et communiquent en co-temporalité tout en partageant les objets de leurs travaux et discussions. Ils se voient attribuer des rôles en relation avec le but de la réunion. L'organisation de leurs interventions est régie par un mécanisme de type "tour de parole".

- La coopération étroite : La coopération étroite correspond à une vision plus précise de la coopération. Elle permet une interaction maximale entre les individus dans un monde cohérent simulant la réalité. Cette forme de coopération se rapproche de la notion de travail coopératif à la planche. C'est notamment dans ce cadre que l'on peut imaginer de nouvelles formes de coopération non encore exploitées. La puissance de cette coopération réside dans la liberté d'action qu'engendre la possibilité d'agir finement et simultanément sur des objets d'un monde virtuel global. Dans ce mode coopération les participants peuvent travailler, communiquer et interagir en temps réel sur tous les objets partagés du projet. Les conséquences de leurs interventions sont directement gérées au niveau des objets manipulés.

5 Analyse des besoins

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. Cette étape consiste à cerner le domaine d'application et le rôle attendu du système tout en prenant en compte les contraintes de développement et d'utilisation.

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 comme le modèle Denver et la méthode GTA pour l'analyse de la tâche.

La modélisation des tâches vient appuyer la phase d'analyse des tâches. L'analyse des tâches est une étape essentielle de la conception de collecticiel, dont le but est d'identifier les actions que peut effectuer l'utilisateur, devant être prises en compte lors de la conception. La compréhension de ces tâches ne peut être obtenue que par une forte implication de l'utilisateur final en observant cet utilisateur sur son lieu de travail ou en observant la façon dont les tâches sont accomplies sur le système interactif courant.

Le résultat de l'analyse des tâches est une liste informelle de tâches pertinentes pour le domaine d'application observé, qui nécessite d'être étudié, afin d'être exploitée lors de la conception du collecticiel. Ainsi, le but de la phase de modélisation des tâches est de construire un modèle qui décrit les intentions des utilisateurs et les activités dans lesquelles ces utilisateurs doivent s'engager pour accomplir leurs buts.

Dans la suite, nous utiliserons le modèle Denver et la méthode GTA en tant que moyens pour caractériser les besoins d'un collecticiel de simulation.

5.1 Modèle de tâche (GTA)

La méthode GTA (Groupware Task Analysis) est une des premières tentatives de modélisation de la complexité des résultats de l'analyse des tâches pour les collecticiels. Cette approche regroupe trois activités :

- Le modèle de tâche 1 est le résultat de l'analyse des tâches courantes. C'est au cours de cette phase que la méthode fournit un cadre de travail, qui présente l'organisation comme composée :
 - de Personnes, car il est nécessaire de faire une distinction entre les acteurs (individus) et les rôles (classes d'acteurs) pour décrire la structure humaine du groupe de travail ;
 - du Travail séparé en un ensemble de tâches organisées en différents niveaux de complexité ;
 - de la Situation, qui décrit les objets, en incluant une analyse de la structure objet, et l'environnement, qui inclut les acteurs et leurs rôles, les conditions pour l'accomplissement des tâches, les objets pertinents et d'autres éléments, comme la technologie de l'information disponible pour la délégation des tâches.
- Le modèle de tâche 2 est la proposition d'un nouveau modèle de tâche. La construction de ce modèle est conduite par la perspective du travail tel qu'il devrait se pratiquer. Ce peut être une réorganisation du modèle 1 ou une refonte complète. Dans tous les cas, il s'agit de résoudre les conflits créés par le premier, en accord avec les souhaits du client.

- La machine virtuelle de l'utilisateur est la spécification de la sémantique du système futur. C'est une description abstraite du système en terme de technologie (contrairement au modèle de tâche 2, qui lui présente le système futur du point de vue de la structure de tâches et organisation du travail).

Pour construire les deux premiers modèles nous avons utilisé l'outil d'analyse des tâches EUTERPE qui a été développée spécialement pour assister les concepteurs pendant le processus d'analyse. Cet outil est basé sur une ontologie dérivée des trois points de vue de la méthode GTA et incorpore des aspects de plusieurs autres méthodes d'analyse de tâches. Cette ontologie décrit la manière avec laquelle nous observons l'environnement de tâche pendant l'analyse par un ensemble de concepts (tâche, objet, rôle, agent, et événement) et de relations (utilise, déclenche, joue, exécuté par, sous-tâche, responsable et utilisé par) entre eux. La Figure 17 représente la construction de la structure hiérarchique des tâches obtenue à l'issue de cette analyse dans l'outil EUTERPE. Dans la suite nous présentons un ensemble de concepts sélectionnés et de relations qui fait partie de cette ontologie :

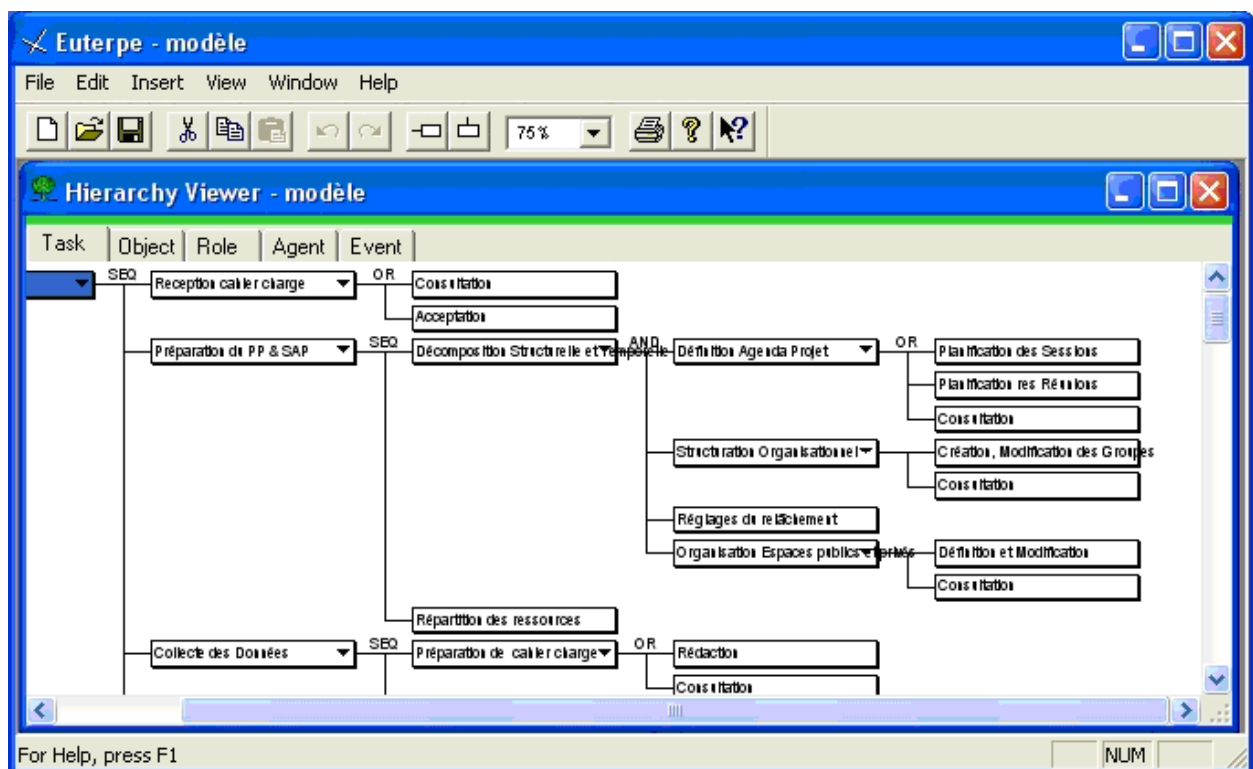


Figure 17 : Construction de la structure de tâche dans l'outil EUTERPE

Les concepts

- *Objet* : Un objet se réfère à une entité physique ou non physique. Il possède des attributs et peut être structuré par une structure de type hiérarchique. Il peut également être inclus dans d'autres objets.
- *Agent* : Un agent est une entité active qui désigne une catégorie d'acteurs avec certaines compétences clairement définies. Cette catégorie se réfère à des individus, des groupes d'individus ou à des logiciels. Au cours du projet, un agent joue un rôle ou éventuellement plusieurs rôles. Le cumul des rôles par un même agent peut être dû, soit à la nécessité de son intervention à plusieurs phases du projet, soit à un contexte particulier d'étude nécessitant le regroupement de compétences variées par un même agent.
- *Rôle* : Un rôle est un ensemble cohérent de tâches pouvant être accomplies à un instant déterminé du projet par un ou plusieurs agents (ou même éventuellement accomplies de façon automatique). Le rôle est dit significatif lorsqu'il a un but clair ou bien lorsqu'il permet de distinguer les groupes d'agents. Le rôle est responsable des tâches qu'il comprend et peut être structuré par une structure hiérarchique.
- *Tâche* : Une tâche est une activité accomplie par des agents pour atteindre un certain objectif et qui requiert une certaine période de temps pour son exécution qui change typiquement son environnement. Les tâches sont exécutées dans un certain ordre et l'achèvement d'une tâche peut déclencher l'exécution d'une ou de plusieurs autres tâches. Une tâche pourrait aussi être déclenchée par un événement qui s'est produit dans son environnement. Les tâches complexes peuvent être décomposées en sous-tâches. Il est important pour ce concept de distinguer entre les unités de tâches et les tâches de base. Idéalement, une unité de tâche devrait être exécutée seulement par l'exécution d'une ou de plusieurs tâches de base. La relation entre les unités de la tâche peut montrer les problèmes qu'un agent peut rencontrer pour atteindre ses objectifs. Une unité de tâche est la tâche la plus simple qu'un utilisateur veut l'accomplir réellement. Une tâche de base est une tâche pour laquelle un système fournit une fonction simple. Usuellement, les tâches de base sont décomposées en actions d'utilisateur et en opérations de système. Une action d'utilisateur est une action faite par les utilisateurs humains. Celle-ci est significative seulement dans le contexte de ses tâches de base. Une

opération système est une action faite par le système. Elle n'est pas une tâche typique parce que, en tant que telle, elle n'a aucun intérêt pour l'utilisateur.

- *Événement* : Un événement est un changement dans l'état de l'environnement de la tâche à un instant de temps précis. Ce changement peut refléter un changement de valeur des attributs des concepts internes ou externes ; Internes tels que l'objet, la tâche et l'agent ou externes comme la météo ou l'alimentation en électricité. En déclenchant des tâches, les événements influencent l'ordre d'exécution des tâches. Dans un projet de simulation, la réception du cahier de charges par le chef de projet représente l'événement qui déclenche le démarrage du projet.

Les relations

Les concepts définis précédemment sont reliés entre eux de manière spécifique. Nous présentons ci-après les différentes relations entre ces concepts. Pour chaque relation, nous définissons et nous expliquons le schéma de la relation. La figure 16 représente l'ensemble des concepts et des relations.

- *Utilise* : *Utilise* (Tâche, Objet, Action) cette relation spécifie l'objet utilisé pendant l'exécution de la tâche et aussi la manière de son utilisation. Le paramètre action spécifie l'acte exercé sur cet objet. La relation *utilise* change typiquement l'état de l'objet.
- *Déclencher* : La relation *Déclencher* (Tâche/Événement, TâcheDéclenchée, TypeDéclenchement) est la base de la spécification du flux de tâche. Elle spécifie soit, un événement et la tâche déclenchée, ou bien le type de déclenchement d'une tâche. Plusieurs types de déclenchement sont possibles tels que le déclenchement conditionné, parallèle ou séquentielle.
- *Joues* : Chaque agent devrait jouer un ou plusieurs rôles. La relation *Joues* (agent, rôle, rendez-vous) spécifie comment ce rôle a été obtenu. Actuellement, le paramètre *Rendez-vous* peut prendre les valeurs : ASSIGNÉ, DÉLÉGUÉ, EXIGÉ ou SOCIAL.
- *Exécuté_Par* : La relation *Exécuté_Par* (Tâche, Agent / Rôle) spécifie qu'une tâche est exécutée par un agent. Ceci ne signifie pas que cet agent est le seul responsable de la tâche, parce que ceci dépend de son rôle et de sa manière d'obtention. Lorsque la

spécification de l'agent n'est pas pertinente, un rôle peut aussi être spécifié comme l'entité exécutante.

- *Sous -Tâche* : La relation *Sous -Tâche* (tâche, Sous -Tâche) décrit la décomposition de la tâche.
- *Sous- Rôle* : La relation *Sous -Rôle* (Rôle, Sous -Rôle) permet la construction de la structure hiérarchique des rôles. Elle signale si un rôle inclut d'autres rôles responsables des tâches. Lorsque le rôle possède des sous rôles, les responsabilités de la tâche sont ajoutées au rôle.
- *Responsable* : Responsable (rôle, tâche). La relation responsable spécifie la tâche sur laquelle le rôle est responsable.
- *Utilisé_Par* : *Utilisé_Par* (objet, agent/rôle, droit). La relation *Utilisé_Par* indique l'historique d'utilisations d'un objet et les pouvoirs d'un agent ou d'un rôle sur cet objet. Les droits des agents sur les objets peuvent être de nature existentielle (créer et détruire). Elle indique aussi le propriétaire (propriétaire) ou le délai de manipulation des objets (utilise, change).

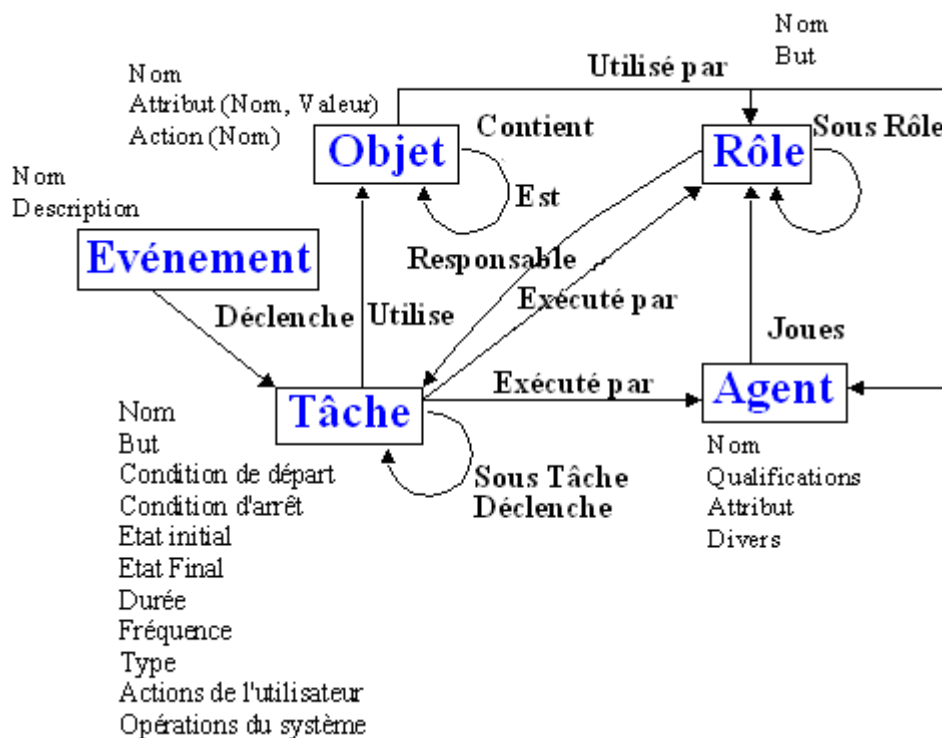


Figure 18 : Concepts et Relations dans GTA (Van der Veer et al., 1996)

5.2 Application de GTA au projet de simulation

Objets

Pour construire un collecticiel, il faut identifier les objets produits ou consommés et faire apparaître les primitives de coopération devant être fournies. Ces objets peuvent être liés au domaine d'étude ou plus générales. Ainsi, l'analyse de la conduite d'un projet de simulation nous à permet d'identifier les principaux objets supports de la coopération. Le Tableau 6 présente un récapitulatif de ces objets et indique aussi les acteurs susceptibles de les utiliser.

Nom	Cahier des charges initial	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Donneur d'ordre	Rédaction
Contient		
Instance		
Support		
Commentaire	Cahier des charges initial du projet	

Nom	Planning prévisionnel & suivi de l'avancement du projet	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Décomposition structurelle et temporelle
	Chef de projet	Répartition des ressources
Contient		
Instance		
Support		
Commentaire		

Nom	Phase ou étape de travail	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Définition, répartition
Contient		
Instance		
Support		
Commentaire		

Nom	Rôles et compétences	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Attribution, modification des droits
	Membres du projet	Consultation rôles et de l'organisation
Contient		
Instance		
Support		
Commentaire		

Nom	Cahier des charges de l'étape	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Rédaction
	Membres du projet	Consultation
Contient		
Instance		
Support		
Commentaire		

Nom	Compte Rendu de l'étape	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Validation
	Membres du projet	Rédaction
	Membres du projet	Consultation
Contient		
Instance		
Support		
Commentaire		

Nom	Objets produits par l'étape	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Membres projet qualifiés	Création et modification
	Chef de projet	Consolidation
	Membres projet autorisés	Consultation et recherche
	Membres projet autorisés	Partage, désignation
	Membres projet autorisés	Sous-Traitance (délégation)
	Membres projet autorisés	Co-Traitance (modification + partage)
Contient		
Instance		
Support		
Commentaire	(Modèles, Résultats, documentations, Rapport...)	

Nom	Agenda de projet	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Planification des sessions et réunions
	Membres du projet	Consultation
Contient		
Instance		
Support		
Commentaire		

Nom	Agenda Individuel	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Membres du projet	Consultation, mise à jour
	Participants Autorisés	Consultation par autre personne
Contient		
Instance		
Support		
Commentaire		

Nom	Session	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Définition
	Membres de la session	Connexion, déconnexion
Contient		
Instance		
Support		

Commentaire	
-------------	--

Nom	Liste des personnels prévus en session	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Définition
	Membres du projet	Consultation
Contient		
Instance		
Support		
Commentaire		

Nom	Liste des personnels présents en session	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Définition
	Membres de la session	Consultation
Contient		
Instance		
Support		
Commentaire		

Nom	Réunion	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Animateur	Définition de l'ordre du jour, animation
	Membres de la réunion	Participation normale ou en retard
	Chef ou membres réunion	Prise de décision (avec ou sans vote)
Contient		
Instance		
Support		
Commentaire		

Nom	Paramètres du WYSIWIS	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Chef de projet	Réglages du relâchement
Contient		
Instance		
Support		
Commentaire		

Nom	Espaces publics et privés	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Participants	Définition
Contient		
Instance		
Support		
Commentaire		

Nom	Stratégies de contrôle	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Gestionnaire système coopératif ou délégué	Sélections
Contient		
Instance		
Support		
Commentaire		

Nom	Outils coopératifs	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Gestionnaire système coopératif ou délégué	Intégration
Contient		
Instance		
Support		
Commentaire		

Nom	Messages (avec données associées)	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Membres du projet	Consultation, envoi, choix mode de com.
Contient		
Instance		
Support		
Commentaire		

Nom	Signaux Audio / Vidéo	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Membres du projet	Emission, réception, paramétrage
Contient		
Instance		
Support		
Commentaire		

Nom	Conseil	
Attributs	Nom	Valeur
Utilisateurs	Nom	Droits
	Membres du projet	Demande de conseil à un expert
Contient		
Instance		
Support		
Commentaire		

Rôles

Dans le cas d'un projet de simulation, les personnes, qui seront les utilisateurs du collecticiel de simulation sont les membres de l'équipe chargée de la réalisation du projet. Les groupes formés par de tels utilisateurs sont caractérisés par : Une certaine stabilité, hétérogénéité du fait qu'un projet de simulation nécessite des compétences pluridisciplinaires, la taille d'un groupe qui est a priori quelconque dépendant de l'envergure du projet.

L'analyse de la méthode de conduite d'un projet de simulation nous a permis de faire apparaître un certain nombre de rôles génériques. Ces rôles peuvent être liés au domaine de simulation d'étude ou plus généraux :

- Chef de Projet,
- Donneur d'ordre,
- Membres du projet qualifiés (Modélisation & Simulation, Probabilités & Statistiques, Domaine d'étude et Programmation),
- Experts consultants (Simulation, domaine d'étude, etc. ...),
- Responsable de la session,
- Membres de la session,

- Chef de la réunion,
- Animateur de la réunion,
- Membres de la réunion,
- exécutant,
- évaluateur,
- valideur,
- consolideur et conseiller
- Administrateur système.

Pour chaque activité du projet, chaque individu possède au moins un rôle issu de ces rôles. Chaque participant possède un certain degré de liberté dans la conception et la réalisation du travail qui lui est confié, mais il doit connaître le contexte de son intervention, c'est-à-dire les liaisons d'interdépendance avec les autres sous-ensembles et avoir une vue générale du projet pour situer les tâches des autres participants. L'attribution des rôles reflète l'organisation du groupe dans un contexte de travail particulier.

Dans le cadre d'un projet de simulation, le chef de projet a le rôle d'animateur. A ce titre, il se porte garant de la cohérence et du bon avancement du projet. Après avoir décomposé l'ensemble à réaliser en un certain nombre de sous-ensembles, il identifie les droits et devoirs qui incombent aux différents participants auxquels il confie l'étude détaillée et la réalisation de chaque phase. Il donne ensuite les contraintes, et suit les travaux tout en arbitrant les éventuels conflits entre participants. Pour finir, il valide les travaux des participants en s'assurant qu'ils sont aptes à réaliser exactement les tâches définies par leurs spécifications. Si tel est le cas, il place les travaux dans un environnement de référence protégé. De ce fait, il assure l'unicité des versions du travail du groupe. Chaque participant peut néanmoins archiver, dans un environnement privé, une version de ses travaux différente de celle retenue par l'animateur.

Agents

Un agent désigne une catégorie d'acteurs avec certaines compétences clairement définies. On repère pour tout projet un certain nombre d'agents génériques. Ces agents peuvent être liés au domaine de simulation d'étude ou plus généraux :

- Client d'étude,
- Ingénieur spécialiste en simulation,

- Analyste,
- Technicien programmeur,
- Agent de saisie,
- Ingénieur spécialiste en domaine d'étude,
- Agents de maîtrise du domaine,
- Consultants en simulation,
- Consultants en domaine d'étude,
- Mathématiciens,
- Spécialiste en réseaux.
- Collecticiel,
- Simulateur.

Arbre des tâches

En appliquant la méthode GTA, nous avons construis le modèle des tâches d'un collecticiel de simulation. La Figure 19 représente la structure hiérarchique des tâches.

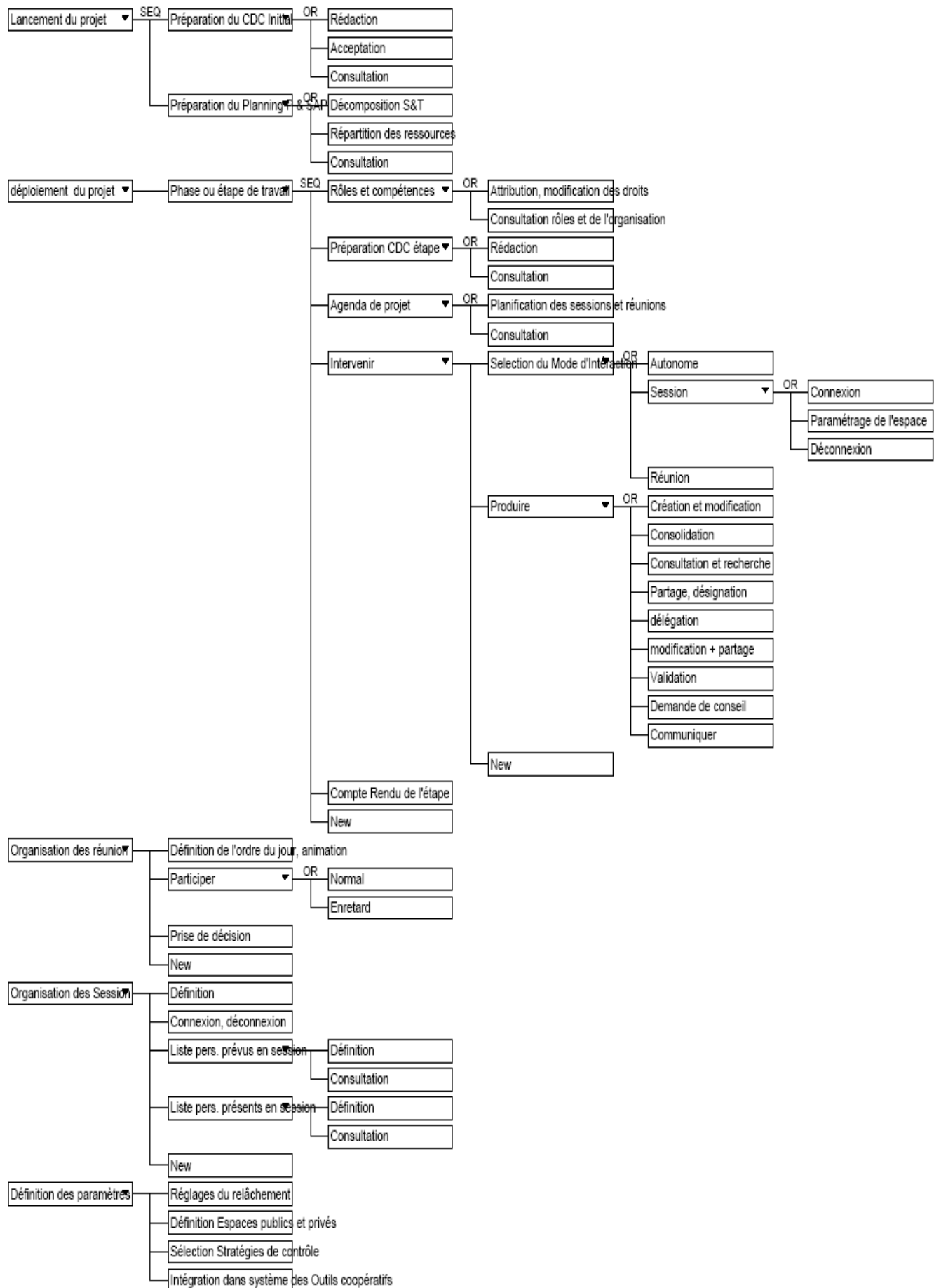


Figure 19 : Arbre des tâches

6 Le modèle de Denver

Bien que le modèle de Denver 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

En appliquant le modèle Denver, nous allons caractériser les quatre modes de coopérations présentés précédemment (Asynchrone, session, réunion et étroite), du point de vue des interactions, situations et protocoles.

6.1 Situation d'interaction

Chaque polygone de la Figure 20 représente un mode de coopération : asynchrone, en session, en réunion et étroite. Les situations d'interaction explicitent ici les différences entre les modes. Toutes les situations ont en commun la taille du groupe et la localisation des membres (même ou des lieux différents). La différenciation se fait au niveau des trois axes "Coordination", "Temporalité" et "Dépendance"

- **La taille d'un groupe** : celle-ci peut être grande ou petite.
- **Le lieu** : l'action peut se dérouler soit dans un même lieu, soit dans des lieux différents.
- **Coordination** : les deux modes réunion (Figure 20 (3)) et étroite (Figure 20 (4)) correspondent exactement à celles qui existent dans le cadre d'un projet (réunion et travail à la planche), c'est-à-dire planifiée. A l'opposé, un mode de coopération asynchrone (Figure 20 (1)) correspond au mode de travail autonome est spontané, c'est une réponse à une requête d'un autre membre ou est une offre spontanée à un membre. Entre les deux, le mode de coopération en session (Figure 20 (2)) est semi-planifiée : c.-à-d. planifiée de point de vue présence uniquement.
- **Dépendance** : la dépendance la plus forte est celle de la coopération étroite (Figure 20 (4)). C'est le chef de projet qui décide quoi faire. En fait, la production d'un objet est ici réalisée par la démarche du chef. A l'opposé les deux modes de coopération asynchrone (Figure 20 (1)) et en session (Figure 20 (2)) la dépendance est relâché; tout le monde

travaille de façon équivalente et indépendante. Pour le mode de coopération en réunion (Figure 20 (3)) la dépendance est possible.

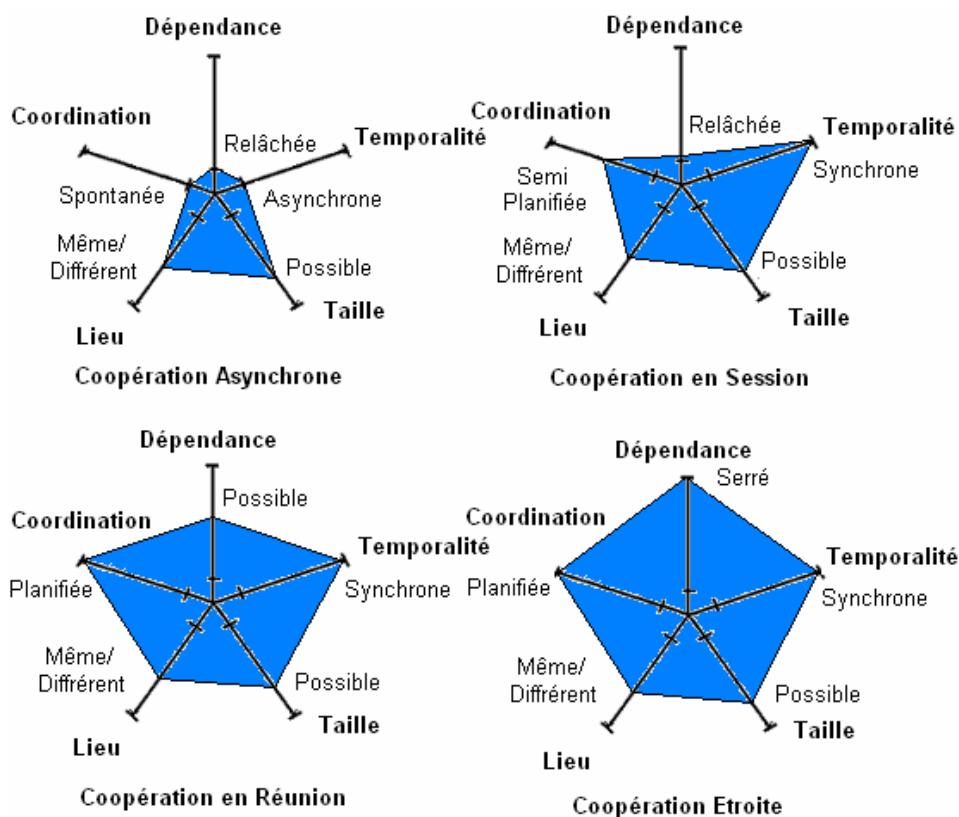


Figure 20 : Les situations d'interaction pour les quatre modes de coopérations

- **Temporalité** : les trois modes de coopération en session, en réunion et étroite (Figure 20 (2), (3), (4)) exigent la connexion ou la présence des membres, c'est-à-dire synchrones à l'opposé du mode de coopération asynchrone (Figure 20 (1)).

6.2 Protocole social de l'interaction

En ce qui concerne les protocoles d'interaction, Chaque polygone de la Figure 21 représente un mode coopération : asynchrone, en session, en réunion et étroite. Tous ont en commun la caractéristique taille du groupe. La différenciation se fait au niveau des trois autres axes.

- **La taille d'un groupe** : celle-ci peut être grande ou petite.
- **Le lieu** : l'action peut se dérouler soit dans un même lieu, soit dans des lieux différents.

- **Conflits et résolution** : Les deux modes de coopération asynchrone (Figure 21 (1)) et en session (Figure 21 (2)) sont les modes de travail autonome, dans ces modes les membres sont libres pour résoudre d'éventuels conflits, ceci justifie alors la valeur basse attribuée à cette caractéristique. A l'opposé le mode coopération étroite (Figure 21 (4)). Pour le mode de coopération en réunion (Figure 21 (3)) la valeur attribuée à cette caractéristique est possible, ceci se justifie par le fait que ce mode correspond exacte aux réunions traditionnelles qui ne comporte pas des situations de concurrence ou d'opposition serrées.
- **Style de réunion** : la valeur attribuée à cette caractéristique dans les deux modes en réunion (Figure 21 (3)) et étroite (Figure 21 (4)) est unidirectionnelle, ceci se justifie par le fait que ces modes sont contrôlés ou animés par le chef de projet et gérés par un style de type tour de rôle. A l'opposé les autres modes asynchrone (Figure 21 (1)) et session (Figure 21 (2)) est multidirectionnelles.
- **Formalise d'échange** : Dans les deux modes asynchrone (Figure 21 (1)) et en session (Figure 21 (2)) les différents participants travaillent de façon autonome, ceci suppose l'existence de certaines formalités de communication entre eux. Dans les autres modes la communication entre les participants est moins formelle par exemple un courrier électronique est plus formel qu'une note écrite sur un Post-It.
- **Réglementation des interventions** : Dans les deux modes de coopération asynchrone (Figure 21 (1)) et en session (Figure 21 (2)) la valeur attribuée à cette caractéristique est la plus basse pour exprimer le mode de travail autonome. A l'opposé le mode de coopération étroite (Figure 21 (4)). Entre les deux, le mode de coopération en réunion. La valeur attribuée à cette caractéristique est possible, ceci exprime certaine liberté d'intervention tolérée aux participants.

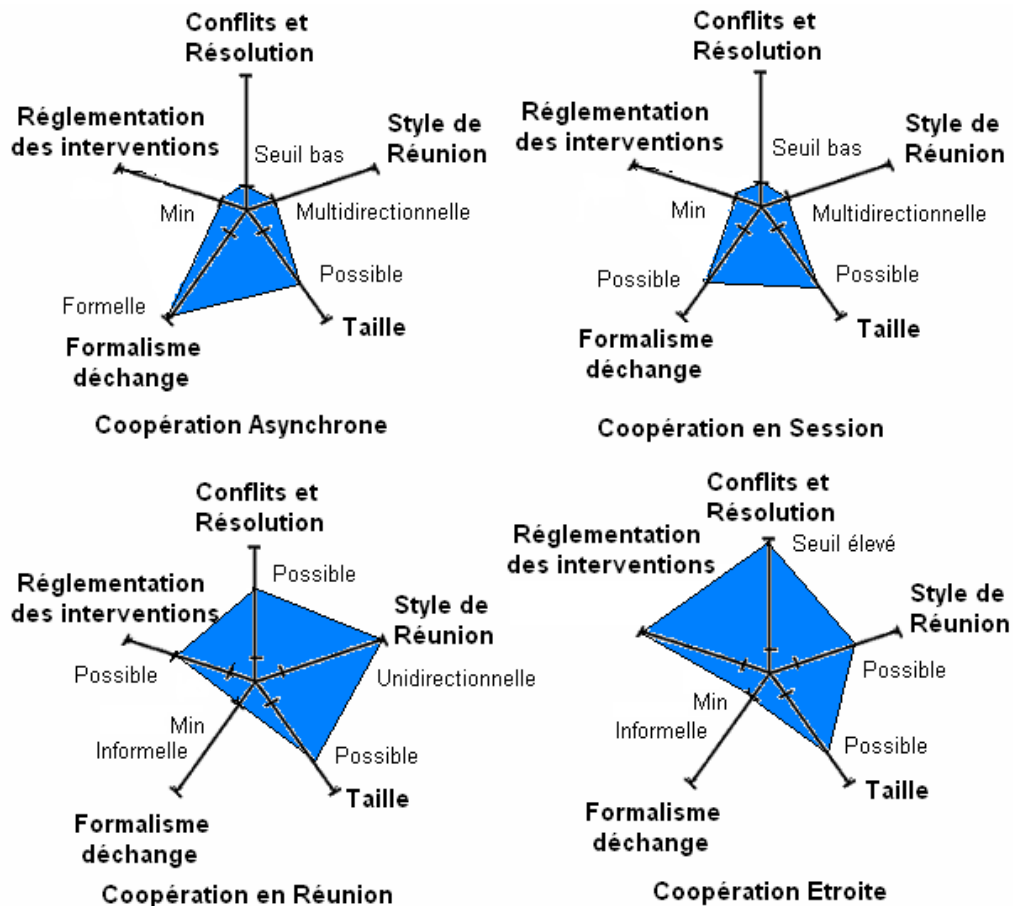


Figure 21 : Les protocoles d'interaction des quatre modes de coopérations

L'utilisation du modèle de Denver nous a donc servi à caractériser un collecticiel de simulation.

7 Spécifications fonctionnelles

Les besoins fonctionnels d'un collecticiel, mêmes lorsque son champ d'application est déterminé, ne sont pas une donnée propre à ce domaine. Ils restent à découvrir et à justifier. Malheureusement dans le domaine de simulation il n'y a pas des travaux qui traitent ce point, mais ceci n'exclut pas la possibilité de s'inspirer des travaux de recherche que nous estimons semblable au nôtre. La liste des besoins que nous exposons ici n'a ni la prétention d'être définitive, ni irréfutable.

1. Schémas-types de suivi de projet. Fonctionnalité permettant le suivi l'avancement du projet. Le collecticiel doit gérer les étapes du processus automatiquement et par exemple, envoyer des rappels aux membres lors de dépassement des délais. Au delà encore de ce besoin, on peut voir un besoin d'explicitation et de transparence des processus.

2. Gestion off-line d'un projet sans être obligé de travailler sur le serveur. Il doit être possible de consulter la base d'information sans se connecter au service. Cela implique que cette base est répliquée localement sur chaque poste. Cette fonctionnalité est révélatrice d'une préoccupation concernant le coût de connexion et indirectement, la vitesse des transferts de données par internet. Ce problème est moins sensible lorsque l'utilisateur dispose par exemple d'une ligne ADSL).
3. Gestion des décisions à prendre ou des points en suspens. Il s'agit de proposer aux utilisateurs des profils-types de résolution de problème, hypothèse faite qu'il existe des problèmes classiques. Autre application : la gestion des comptes-rendus de réunions. Chaque point de réunion étant clairement explicite, le système montre aux participants les points qui ont été réglés depuis la dernière réunion et les points qu'ils restent à résoudre ; la diffusion du compte-rendu est automatisée.
4. Outil de partage d'applications. Afin de pouvoir travailler en mode de coopération étroite, il est nécessaire que le collecticiel dispose de possibilité de partage d'application. Une autre application serait l'édition conjointe des documents.
5. Informations sur les actions accomplies sur les éléments de base. Il y a un besoin de traçabilité des objets ; " qui a déposé, chargé, modifié un objet, et quand ? " sur les questions auxquelles devraient pouvoir répondre un collecticiel.
6. Gestion de profils des utilisateurs. Le profil d'utilisateur dont il est question ici est la capacité de l'utilisateur à lire des documents sous différents formats. Ce profil, attaché à la fiche signalétique des utilisateurs, permet à l'émetteur d'un fichier de savoir si son récepteur est capable de " relire " le fichier, pour s'en servir ou le modifier. Si ce n'est pas le cas, il pourra choisir un autre format " passerelle " en fonction de ce que lui indiquera le profil de son correspondant.
7. Routage d'un document vers un tiers. Il arrive qu'un membre veuille communiquer un document à une personne qui n'est pas référencée dans le collecticiel. Il faudrait alors pouvoir lui transmettre ce document, par exemple en indiquant l'adresse électronique de cette personne.
8. Annuaire des objets produit à jour. Lorsqu'un objet est modifié et mis à jour dans la base, il est utile que les utilisateurs concernés en soient avertis. Plus généralement, il serait

opportun de proposer une fonction qui présenterait toutes les mises à jour (depuis la dernière synchronisation). Le routage d'objets ne peut être compris et justifié qu'à partir du moment où l'on considère que l'accès du collecticiel est réservé aux personnes qui sont enregistrées auprès du service. Cette fonctionnalité est relative à ce que l'on appelle dans le domaine du TCAO la conscience de groupe, et particulièrement la conscience de l'espace de travail (workspace awareness).

7.1 Primitives de coopération

A partir de l'analyse de besoins, il convient alors de concevoir le modèle de l'interaction. Les spécifications externes d'un collecticiel s'expriment en termes de requis fonctionnels et requis utilisateurs. Les requis fonctionnels cernent les fonctionnalités offertes par le collecticiel. Le modèle du trèfle, présenté dans le chapitre précédent, fournit un cadre conceptuel utile pour organiser et définir les fonctionnalités d'un collecticiel. En effet, pour construire un collecticiel, il faut identifier les objets produits ou consommés et faire apparaître les primitives de coopération devant être fournies. Ces objets peuvent être liés au domaine d'étude ou plus générales. Ainsi, l'analyse des besoins d'un collecticiel de simulation présenté précédemment nous a permis d'identifier les principaux objets supports de la coopération. Le Tableau 7 présente un récapitulatif de ces objets et indique aussi les acteurs susceptibles de les utiliser.

Objets	Primitives de coopération	E	Acteurs
Cahier des charges initial	Rédaction	P	Donneur d'ordre
Planning prévisionnel & suivi de l'avancement du projet	Décomposition structurelle et temporelle	R	Chef de projet
	Répartition des ressources	R	Chef de projet
Phase ou étape de travail	Définition, répartition	R	Chef de projet
Rôles et compétences	Attribution, modification des droits	R	Chef de projet
	Consultation rôles et de l'organisation	R	Membres du projet
Cahier des charges de l'étape	Rédaction	R	Chef de projet
	Consultation	R	Membres du projet
Compte Rendu de l'étape	Rédaction	R	Membres du projet
	Validation	R	Chef de projet
	Consultation	R	Membres du projet
Objets produits par l'étape (Modèles, Résultats, documentations, Rapport...)	Création et modification	P	Membres projet qualifiés
	Consolidation	P	Chef de projet
	Consultation et recherche	P	Membres projet autorisés
	Partage, désignation	P	Membres projet autorisés
	Sous-Traitance (délégation)	P	Membres projet autorisés
	Co-Traitance (modification + partage)	P	Membres projet autorisés
Agenda de projet	Planification des sessions et réunions	R	Chef de projet
	Consultation	R	Membres du projet
Agenda Individuel	Consultation, mise à jour	P	Membres du projet
	Consultation autre personne	P	Participants Autorisés
Session	Définition	R	Chef de projet
	Connexion, déconnexion	R	Membres de la session
Liste des personnels prévus en session	Définition	R	Chef de projet
	Consultation	R	Membres du projet
Liste des personnels présents en session	Définition	R	Automatique
	Consultation	R	Membres de la session
Réunion	Définition de l'ordre du jour, animation	R	Chef de projet-Animateur
	Participation normale ou en retard	R	Membres de la réunion
	Prise de décision (avec ou sans vote)	R	Chef ou membres réunion
Paramètres du WYSIWIS	Réglages du relâchement	P	Chef de projet
Espaces publics et privés	Définition	P	Participants
Stratégies de contrôle	Sélection	C	Gestionnaire système coopératif ou délégué
Outils coopératifs	Intégration dans système	P	
Messages (avec données associées)	Consultation, envoi, choix mode de com.	C	Membres du projet
Signaux Audio / Vidéo	Emission, réception, paramétrage	C	Membres du projet
Conseil	Demande de conseil à expert	C	Membres du projet
Simulateur	Exécution, Partage	P	Membres du projet

E : Espace fonctionnel, P : Production, R : Coordination et C : Communication

Tableau 6 : Les objets produits ou consommés et les actions associées

A partir du tableau précédent, on peut rapidement identifier les primitives génériques suivantes : fonctions d'édition (création, destruction, consultation, modification), fonctions de structuration (décomposition, consolidation), validation, diffusion et partage synchrone avec d'autres participants.

A partir des rôles génériques évoqués précédemment on peut proposer la liste des primitives accessibles à un rôle déterminé. Le tableau ci-dessous présente cette liste restreinte aux catégories d'objets les plus représentatives.

Catégorie	Production		Com.	Coordination			
Objets Primitives	Objets de Simulation	Cahier des charges	Messages A/V	Tables de droits	Tables individus	Listes de membres	Agendas, Etapes
Création	MP	DO	MP	AS	AS	R	CP
Destruction	MPA	DO	MPA	AS	AS	R	CP
Consultation	MPA	MPA	MPA	MPA	MP	MP	MPA
Modification	MPA	CP	MPA	CP	AS	R	CP
Décomposition	MPA	CP	MPA	AS	AS	R	CP
Consolidation	CP	CP	MPA	AS	AS	R	CP
Validation	CP	CP	MPA	CP	AS	R	CP
Diffusion	MPA	CP	MPA	AS	AS	R	CP
Partage	MPA						

DO : Donneur d'ordre, CP: Chef de projet, AS : Administrateur système, MP : Membres du projet, MPA : Membres du projet autorisés, R : Responsable réunion ou session.

Tableau 7: Rôles en fonction des données et primitives de coopération

Dans le tableau précédent, nous nous sommes attachés à faire correspondre les objets et primitives aux trois espaces fonctionnels d'un collecticiel (production, communication et coordination). En fait, nous pouvons regrouper l'ensemble des données manipulées en quatre groupes, suivant le type d'activité qu'elles supportent. A savoir : les activités liées aux tâches du domaine de simulation, les activités de réglage de l'interaction entre les participants (paramètres de WYSIWIS, politiques de contrôle), les activités de construction des modes de travail (sessions et réunions) et enfin les activités d'interaction avec l'environnement (agenda, importation et exportation). Cependant, un cinquième groupe de données peut être défini si l'on considère distinctement les activités de workflow. Dans cette catégorie, on retrouve les phases ou étapes de travail, les modélisations des ressources disponibles, etc.

8 Modèle d'architecture logiciel

8.1 Production d'une architecture

Le modèle d'architecture constitue un des éléments indispensables du triplet (Méthode, Modèle, Outils) sur lequel s'appuie le génie logiciel moderne. Il systématise à la fois l'élaboration et l'utilisation des logiciels, et joue un rôle primordial dans leur qualité et leur efficacité. L'utilisation de modèles d'architecture logicielle peut avoir des effets bénéfiques sur au moins cinq aspects du développement des logiciels: La compréhension, la réutilisation, l'évolution, l'analyse et la gestion.

Un modèle d'architecture doit satisfaire trois objectifs. Il vise tout d'abord l'organisation de la mise en œuvre, c'est-à-dire le découpage en modules pour faciliter l'implémentation, la portabilité et la maintenance. Il joue également un rôle plus important en exprimant le découpage fonctionnel utile non seulement lors de la mise en œuvre mais également pendant la démarche d'analyse et de conception. Le troisième rôle du modèle d'architecture est celui d'aide à l'utilisation. Frank-Trapin (Tarpin-Bernard, 1997b) considère qu'un modèle d'architecture constitue le pivot des activités d'élaboration (spécification, conception et réalisation), et qu'en plus, il doit pouvoir guider l'utilisateur dans sa compréhension du mode de fonctionnement des applications. C'est pour cela il donne au modèle d'architecture le triple objectif suivant : servir de support lors de la spécification, constituer l'ossature de la réalisation, et servir de guide d'utilisation.

Une architecture logicielle est assimilée à un ensemble organisé de composants dont les interactions sont médiatisées par des entités spécialisées ou connecteurs. Le processus de conception d'une architecture logicielle, comme le montre la Figure 22, recouvre les deux principales activités suivantes : définition de la décomposition fonctionnelle du système, choix d'une organisation structurelle (vue statique et dynamique). Le résultat de cette activité constitue la structure modulaire ou architecture conceptuelle :

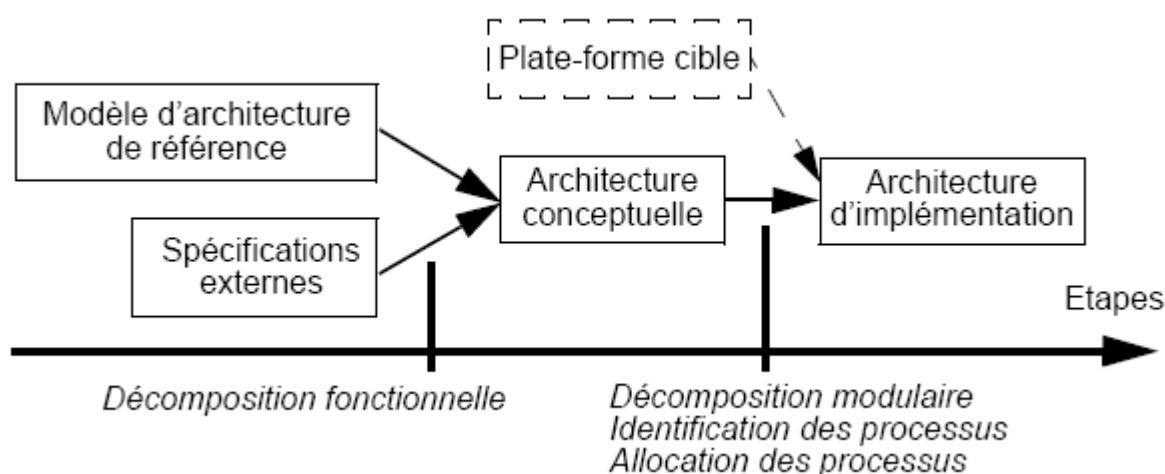


Figure 22 : Etapes de production d'une architecture (Laurillau, 2002)

- Définition d'une décomposition fonctionnelle : cette activité consiste à identifier les fonctionnalités du système regroupées sous forme d'entités logicielles ou de composants. Cette étape est complexe puisqu'il s'agit d'exprimer les requis utilisateurs et les spécifications externes de l'interface en termes logiciels.
- Choix d'une organisation structurelle : il s'agit de choisir ou de définir un schéma d'organisation des différents composants identifiés lors de la décomposition fonctionnelle,

ainsi qu'un protocole d'échanges entre ces composants. Ce protocole est défini par l'ensemble des connecteurs et des règles définissant l'interaction entre les composants. L'organisation obtenue reflète une vue statique (agencement des composants) et dynamique (protocole d'échanges et migration fonctionnelle) du système.

Une fois la structure modulaire du logiciel établie, il convient de focaliser sur l'association entre entités structurelles et processus d'exécution du système, voire l'allocation des processus aux processeurs. Cette dernière activité prend tout son sens pour un collecticiel, système par définition réparti. La structure physique est alors conçue : elle complète la structure modulaire de l'étape précédente pour constituer ensemble l'architecture implémentationnelle. Cette architecture décrit l'organisation des fonctionnalités selon un ensemble de modules qui représentent les unités logicielles qui seront développées. De plus, cette architecture traduit l'aspect dynamique du logiciel en détaillant la répartition du logiciel selon les différents processus. De plus, comme le montre la Figure 19, le développement d'un collecticiel semble de plus en plus dépendant du choix de la plate-forme cible car les choix techniques ont un impact direct sur son développement (ressources disponibles en matière de réseau ou de capacités calculatoires, etc.).

8.2 Modèles de références

Pour élaborer un modèle d'architecture logicielle, les concepteurs peuvent s'inspirer de modèles d'architecture de référence. Ces modèles offrent une description standard d'une organisation de composants et de règles régissant les relations entre ces composants.

Les modèles classiques

Le modèle triple centralisé, répliqué ou hybride fait quasiment office de modèle de référence dans la littérature sur les architectures des systèmes multi-utilisateurs.

Dans le cas centralisé, un processus unique s'exécute sur un site serveur et gère l'ensemble de l'application multi-utilisateur, c'est-à-dire le noyau fonctionnel et l'ensemble des interfaces des utilisateurs. Ce modèle est bien adapté à un système graphique client-serveur comme X-Window. La figure 23 présente une extension de ce modèle dans laquelle différentes interfaces correspondant à différents rôles sont pris en compte.

Le modèle centralisé présente l'avantage de la simplicité : comme l'application est réduite à un seul processus, les questions de communication et de synchronisation entre processus ne se posent

pas. Or elles constituent une difficulté de la conception logicielle des systèmes multi-utilisateurs. Mais cette simplicité se paie en termes de fiabilité et de flexibilité de l'architecture. Le problème de fiabilité est connu : comme toute l'architecture repose sur un seul processus, sa défaillance ou son manque de performance ont des conséquences gênantes pour l'ensemble des utilisateurs. D'autre part, un processus central ne permet pas de garantir pour tous les utilisateurs un temps de réponse acceptable ni même constant. Les propriétés de stabilité et de conformité du temps de réponse sont compromises. Cette architecture présente aussi l'inconvénient de ne pas prendre en compte tous les types de systèmes. Un système qui requiert un parallélisme est limité par une architecture centralisée.

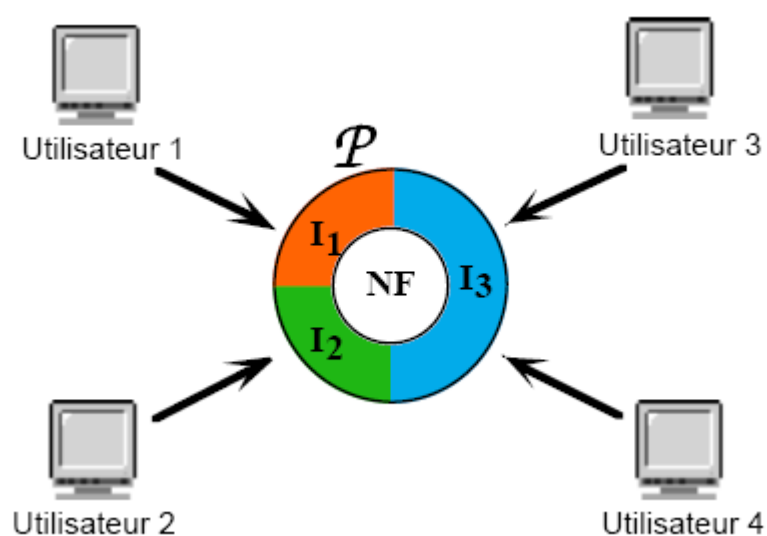


Figure 23 : Le modèle centralisé (Salber, 1995)

La figure précédente représente un exemple de configuration avec quatre utilisateurs, quatre terminaux et un serveur. Les flèches signifient “est client de”. P est l'unique processus du système. Il gère le noyau fonctionnel NF et une ou plusieurs interfaces utilisateur $\{I_i\}$. Une interface I_i présente et permet d'utiliser les services correspondant au rôle i . Sur la figure, les utilisateurs 3 et 4 ont le même rôle. Dans le modèle centralisé originel, $i = 1$

Le modèle répliqué est une perspective radicalement opposée. Cette fois, un processus est alloué à chaque utilisateur et les éventuelles données partagées sont dupliquées sur tous les sites. La figure 24 présente une généralisation du modèle répliqué qui prend en compte différents rôles.

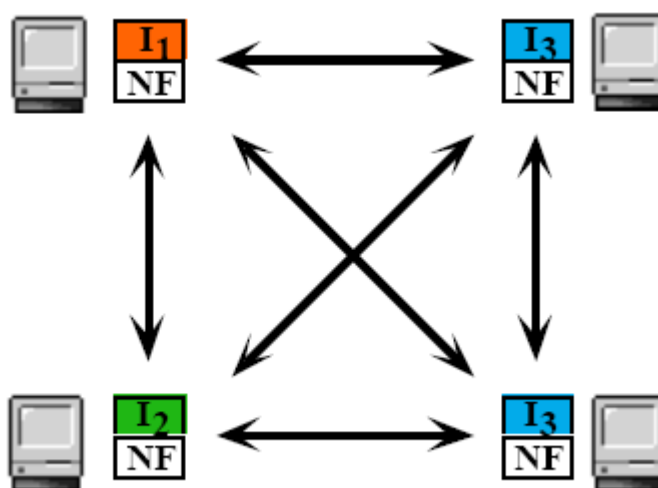


Figure 24 : Le modèle répliqué (Salber,1995)

La figure 24 représente un exemple de configuration à quatre utilisateurs. Un processus par utilisateur gère une copie du noyau fonctionnel NF répliqué et l'interface locale I_i adaptée au rôle de l'utilisateur. Chaque processus peut communiquer directement avec n'importe quel autre. Les connecteurs représentés par des flèches sont un protocole réseau de communication inter-applications. Ils relient deux à deux les noyaux fonctionnels et les interfaces

Ce modèle semble a priori mieux adapté aux systèmes multi-utilisateurs : la duplication des composants de l'architecture permet d'améliorer la vivacité de l'interface. La duplication permet aussi une fiabilité accrue : en cas de défaillance de l'un des processus, les autres processus continuent leur activité. Au cas où le processus défaillant doit communiquer avec d'autres processus, des techniques classiques (communication asynchrone, délai de garde ou "timeout") permettent de détecter la défaillance et de mettre "hors circuit" le processus fautif. Cependant, l'architecture répliquée, si elle est bien adaptée aux services de la classe communicateurs, présente l'inconvénient de dupliquer les dépositaires. Les données étant répliquées sur tous les sites, il devient indispensable de mettre en place des mécanismes de maintien de la cohérence des données répliquées. Cette approche présente les inconvénients suivants.

- Les communications n'étant pas instantanées, des phénomènes de "zone grise" peuvent se produire : la modification d'une donnée sur un site ne sera répercutée sur les autres sites qu'après un intervalle de temps Δt dépendant du réseau de communication. Un réseau de communication (tel Ethernet) étant non déterministe, Δt est en général imprévisible. Des conflits peuvent donc apparaître si plusieurs utilisateurs modifient la même donnée simultanément. Pour résoudre ce problème, des protocoles de

communication garantissant l'ordonnement global des messages, tels les protocoles multicast sont envisageables. L'inconvénient de cette technique est qu'un protocole de ce type a un temps de réponse qui croît exponentiellement en fonction du nombre de sites. Cette solution n'est donc pas envisageable pour un grand nombre de sites. D'autres protocoles, comme ceux fondés sur des algorithmes optimistes, sont toutefois des solutions moins coûteuses et prometteuses.

- En cas de défaillance d'un processus ou du réseau, l'intégrité des données locales au processus fautif (ou de l'ensemble des données dans le cas du réseau) est compromise. Il est indispensable d'envisager des mécanismes robustes de reprise après panne. Ces mécanismes peuvent être analogues aux mécanismes de fusion de versions de documents et nécessitent souvent dans ce cas l'intervention des utilisateurs pour résoudre les conflits. D'autres solutions reposant sur l'utilisation d'un historique des modifications sont aussi utilisées, mais requièrent le maintien d'un historique local à chaque processus.

Le modèle hybride apporte une solution au problème du maintien de la cohérence des données sans sacrifier la vivacité du temps de réponse de l'interface. Cette approche combine les deux cas précédents en décomposant un même système en deux parties, l'une centralisée, l'autre répliquée. La figure 25 présente le modèle hybride étendu au cas d'un système à plusieurs rôles.

Cette solution permet d'isoler dans la partie centralisée les données partagées et conserve le modèle répliqué pour la gestion locale. Notons qu'il n'est pas nécessaire de centraliser les données partagées. On peut utiliser le composant centralisé comme un simple synchronisateur et garder des dépositaires locaux à chaque processus utilisateur. Le synchronisateur central est alors un "chef d'orchestre" qui gère le contrôle d'accès aux données partagées répliquées. Cette solution garantit de meilleurs temps de réponse du système lorsqu'un utilisateur accède aux données partagées. Mais en contrepartie, les données partagées doivent être maintenues cohérentes.

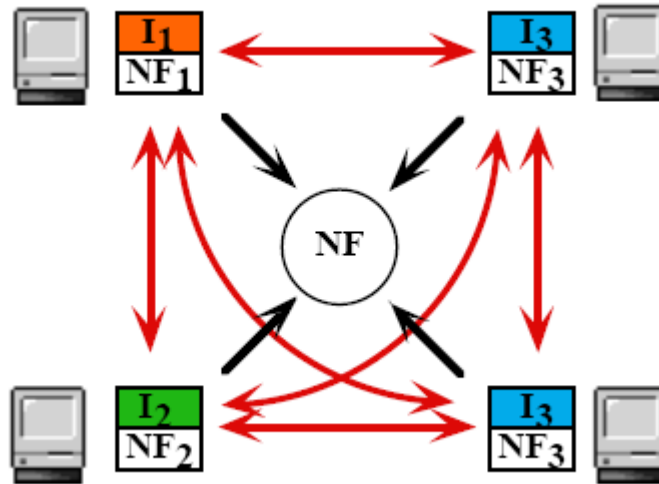


Figure 25 : Le modèle hybride (Salber,1995)

La figure 25 représente un exemple de configuration à quatre utilisateurs. Un processus gère la station de chaque utilisateur (noyau fonctionnel local NF_i et interface locale I_i) et un processus central gère le noyau fonctionnel commun NF . Les communications entre les processus combinent les possibilités du cas centralisé avec le processus central et celles du cas répliqué pour les processus utilisateurs.

Les modèles monolithiques

Le modèle de zipper

Le modèle Zipper repose sur la notion d'états partagés. Ainsi, un collecticiel est décomposé, comme le montre la Figure 26, selon quatre niveaux d'états qui définissent des niveaux d'abstraction :

- Etat de l'écran (Display) qui correspond à l'état des périphériques d'entrée et de sortie (moniteur, souris, etc.),
- Etat de la vue (View) qui correspond à l'état de la présentation logique des données, c'est-à-dire l'état de l'interface utilisateur,
- Etat du modèle (Model) qui correspond au noyau fonctionnel et aux objets du domaine,
- Etat du fichier (File) qui correspond à la représentation persistante du modèle.

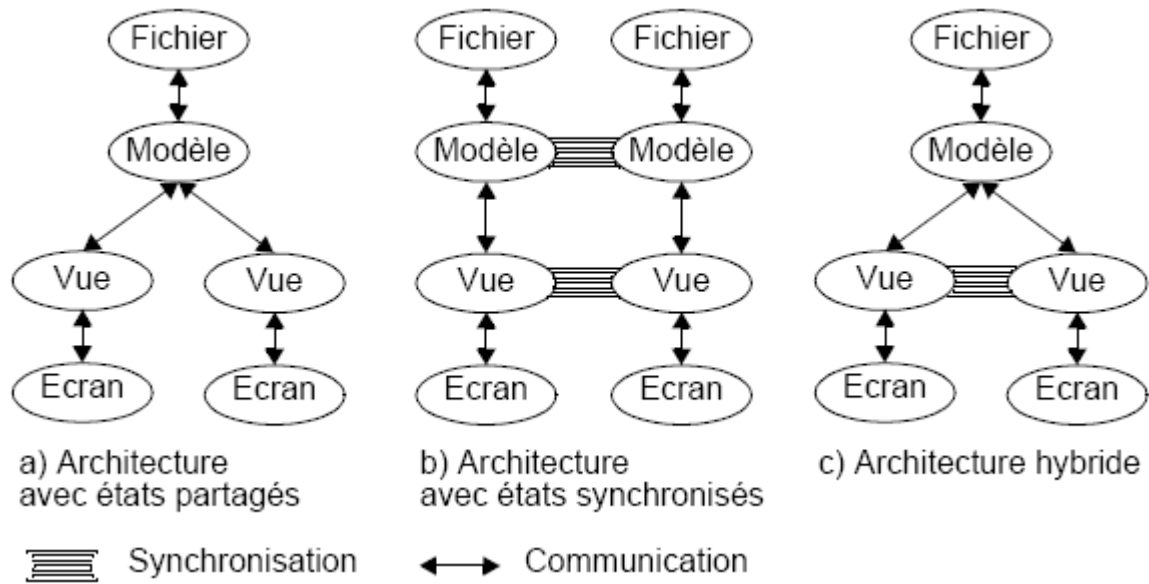


Figure 26 : Différentes formes du modèle Zipper (Laurillau, 2002)

Ces états peuvent être instanciés selon trois modes : partagé, synchronisé ou répliqué. Dans le mode partagé, un état est instancié sous forme unique. Dans le mode répliqué, il existe autant d'instances d'un état qu'il y a d'utilisateurs. Le mode synchronisé est similaire au mode répliqué, mais un mécanisme de synchronisation est chargé de maintenir la cohérence entre toutes les copies (états identiques). La synchronisation d'états répliqués a été introduite pour rendre les collecticiels plus performants puisque les états synchronisés contiennent exactement la même information mais avec une mise à jour plus ou moins retardée en fonction du degré de couplage. Une architecture hybride est constituée à la fois d'états partagés, répliqués et synchronisés. L'exemple de la Figure 26 (a) représente une architecture constituée de deux états partagés (le fichier et le modèle) et de deux états répliqués. L'exemple de la Figure 26 (b) représente une architecture répartie avec synchronisation de deux états (le modèle et la vue). Enfin, l'exemple de la Figure 26 (c) représente une architecture hybride, une combinaison des deux exemples précédents, constituée de deux états partagés (le fichier et le modèle) et de deux états répliqués avec une vue synchronisée. La métaphore de la fermeture éclair qui a inspiré le nom du modèle, Zipper, fait écho au niveau de fermeture de l'architecture par des états partagés. Dans l'exemple de la Figure 26 (a), la fermeture est réalisée au niveau de l'état modèle. Au contraire, dans l'exemple de la Figure 26 (b), l'architecture est totalement ouverte.

Le Modèle d'architecture générique (Dewan)

Le modèle d'architecture pour les collecticiels de Dewan repose sur une structuration en couches. Ainsi, un collecticiel est constitué d'un nombre variable de couches représentant plusieurs niveaux d'abstraction. La couche la plus haute (niveau N) est de nature sémantique alors que la couche la plus basse (niveau 0) représente le matériel comme le montre la Figure 27. L'architecture globale est constituée d'une racine et de plusieurs branches. La racine, comme le montre la Figure 19, est composée de couches partagées (niveaux L+1 à N). Les branches sont composées de couches répliquées (niveaux 0 à L), reliées à la racine au niveau de la couche L. De plus, les objets gérés par les branches sont privés. Réciproquement, les objets gérés par les couches partagées sont publics. Les couches communiquent entre elles à l'aide de deux types d'événements : les événements d'interaction (échangés le long de l'empilement des couches) reflétant l'interaction de l'utilisateur avec le système et les événements de la collaboration (échangés entre les couches appartenant à deux branches différentes) reflétant l'interaction entre tous les utilisateurs.

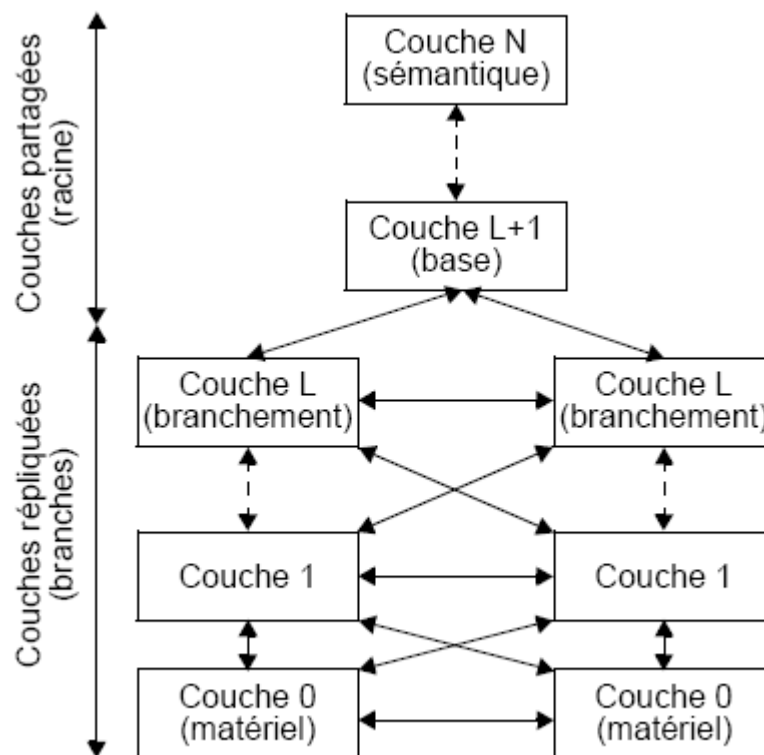


Figure 27 : Modèle d'architecture de Dewan (Laurillau, 2002)

Ce modèle distingue les couches collaboratives de celles qui implémentent uniquement des fonctionnalités mono-utilisateur. A travers cet étiquetage, Dewan propose un moyen pour identifier à quel niveau un collecticiel doit mettre en œuvre la collaboration, à savoir si la collaboration relève de la couche la plus haute (Noyau Fonctionnel) ou si elle relève plus de la Présentation. En ce sens,

le degré de collaboration indique quelle est la couche de plus haut niveau qui est collaborative, sachant que les couches situées à un niveau inférieur peuvent être aussi collaboratives. Ainsi, si le degré est faible, alors la collaboration est mise en œuvre au niveau de la Présentation. Si le degré est fort, alors la collaboration est mise en œuvre au niveau du Noyau Fonctionnel. Ce degré permet de déterminer les couches susceptibles de générer des actions concurrentes et ainsi de déterminer à quel niveau il est nécessaire d'implémenter des mécanismes de contrôle d'accès, de verrouillage des données ou de couplage des données.

Les modèles multi-agent

Le Modèle ALV

Le modèle d'architecture ALV (Abstraction-Link-View), utilisé dans le système RendezVous, vise les collecticiels synchrones construits autour d'un noyau fonctionnel centralisé. Ce composant central est dénommé "Abstraction partagée" dans le modèle. Chaque utilisateur dispose d'un composant interface qui lui est propre et adapté à son rôle ("Vue Personnelle"). Des composants de liaison, "Lien", font le lien entre l'abstraction partagée et chaque vue. Les Link maintiennent des systèmes de contraintes et propagent les modifications de l'un des deux composants vers l'autre. La figure 28 présente une configuration du modèle ALV pour quatre utilisateurs.

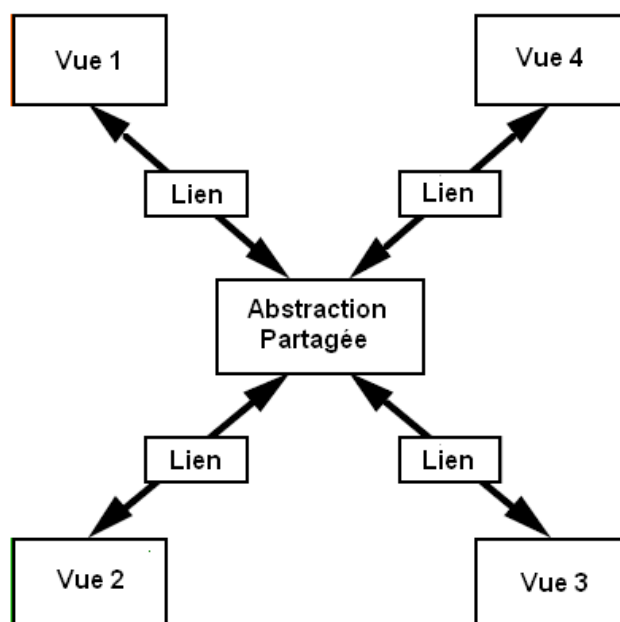


Figure 28 : Le modèle ALV (Salber,1995)

Les flèches représentent des contraintes liant un composant abstraction partagé à des vues propres à chaque utilisateur. Les composants Link maintiennent et propagent les contraintes entre les composants abstraction et vue. Sur la figure quatre utilisateurs partagent une même abstraction

Une architecture selon ce modèle est une organisation d'agents dont les facettes sont les suivantes :

- Une abstraction partagée (Shared Abstraction) qui gère les objets du domaine, partagés par tous les utilisateurs,
- Vue (Personal View) répliquée qui interprète les entrées d'un utilisateur et qui gère les sorties. Les événements générés par l'interaction sont traités au niveau de la vue par des fonctions dédiées qui modifient les données localement,
- Un lien (Link) qui a, d'une part, la charge de relier la facette abstraction avec une facette vue et qui, d'autre part, doit s'assurer que les données locales à une vue sont conformes avec la représentation de la donnée au niveau de l'abstraction.

D'un point de vue global, l'architecture d'un collecticiel élaborée à l'aide du modèle ALV est constituée d'une abstraction partagée reliée à plusieurs vues réparties, une par utilisateur, à l'aide de liens.

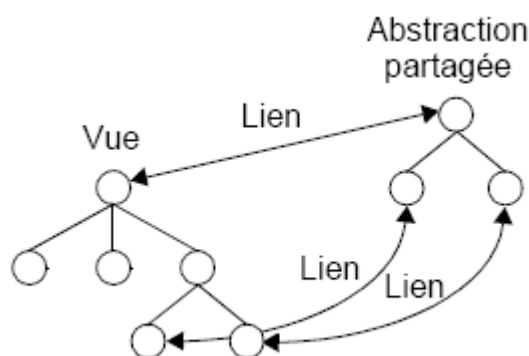


Figure 29 : Les Agents du modèle ALV (Laurillau, 2002)

Ainsi, l'exemple de la Figure 29 représente l'architecture du système à l'exécution avec quatre utilisateurs qui se traduit par l'existence de quatre vues personnelle se partageant l'abstraction à l'aide de quatre liens. A grain plus fin, cette architecture est constituée d'une multitude d'agents ALV dont les facettes Abstraction et Vue sont organisées, comme le montre la Figure 29, sous la forme de deux hiérarchies d'agents qui constituent l'abstraction partagée et la vue globale. Le lien assure alors la mise en correspondance entre les facettes Abstraction et Vue d'un agent. Comme nous pouvons le constater, la vue globale et l'abstraction partagée ne sont pas obligatoirement constituées d'une

hiérarchie identique de facettes. Il est alors possible que certaines facettes vue soient entièrement locales à la partie cliente et indépendantes de l'aspect collaboratif de l'application (non reliés à une facette abstraction).

Enfin, les liens ont la charge d'assurer la cohérence des données entre leur représentation abstraite et la valeur relative à la présentation. De plus, le lien est défini par un ensemble de contraintes traduisant le degré de couplage : selon le choix des contraintes, il est possible de coupler fortement une partie des vues (i.e. un sous-arbre de la hiérarchie) ; au contraire, certaines parties de la vue peuvent être totalement indépendantes.

Les modèles hybrides

Le Modèle PAC*

Le modèle l'architecture PAC*, est une extension du PAC (Présentation Abstraction Contrôleur). Ce dernier est un modèle multi-agent pour la conception logicielle des systèmes interactifs. Il repose sur deux principes directeurs : le concept d'agents réactifs à facettes (système de traitement de l'information) et l'organisation hiérarchique de ces agents. Un agent PAC, comme le montre la Figure 30, est composé de trois facettes :

- Une facette Abstraction (A) qui gère les concepts du domaine et définit la compétence de l'agent indépendamment de la présentation,
- Une facette Présentation (P) qui définit l'interface utilisateur et interprète les entrées et sorties générées par l'utilisateur au cours de l'interaction,
- Une facette Contrôle (C) qui, d'une part, fait le lien entre les facettes Abstraction et Présentation et qui, d'autre part, assure et gère les relations avec les autres agents dans la hiérarchie (l'agent père et les agents fils). Les agents communiquent entre eux uniquement à travers cette facette Contrôle.

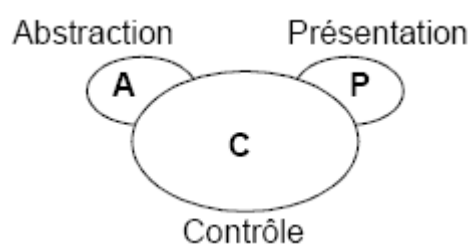


Figure 30 : Un agent PAC (Laurillau, 2002)

Le modèle PAC est une extension du modèle Arch selon une approche multi-agent. Ce modèle reprend les cinq niveaux fonctionnels du modèle Arch et structure le Contrôleur de Dialogue avec une hiérarchie d'agents PAC.

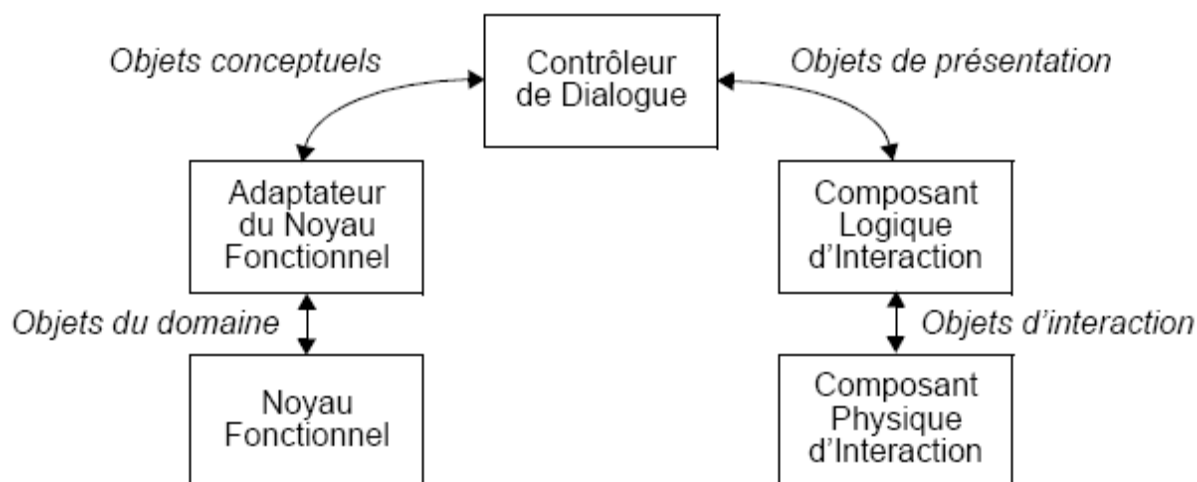


Figure 31 : Modèle de référence Arch (Laurillau, 2002)

Le modèle Arch est un modèle d'architecture qui offre une décomposition canonique des principaux composants d'un système interactif. Le principe véhiculé par le modèle, comme le montre la Figure 31, est de séparer l'interface utilisateur du Noyau Fonctionnel (logique de l'application). En pratique, le Noyau Fonctionnel ne doit avoir aucune connaissance des fonctionnalités relevant de l'interface utilisateur pour faciliter une conception itérative de l'interface, pour favoriser la réutilisation et pour favoriser la portabilité du logiciel. Les composants de ce modèle d'architecture sont :

- Le Noyau Fonctionnel implémente les fonctionnalités et les concepts du domaine indépendamment de leur présentation. Les structures de données manipulées par ce composant sont les objets du domaine.
- L'Adaptateur du Noyau Fonctionnel joue un rôle de médiateur entre le Noyau Fonctionnel et le Contrôleur de Dialogue. Les données échangées avec le Noyau Fonctionnel sont les objets du domaine que le NF exporte vers l'utilisateur. Les données échangées avec le Contrôleur de Dialogue sont des objets conceptuels correspondant à une représentation mentale de l'utilisateur des objets du domaine.
- Le Composant Physique d'Interaction représente les interacteurs logiciels (widget) et matériels. Il s'agit en général d'une boîte à outils graphique (User Interface Toolkit) et des périphériques d'interaction.

- Le Composant Logique d'Interaction joue aussi un rôle de médiateur entre le Contrôleur de Dialogue et le Composant Physique d'Interaction. Ce composant, généralement assimilé à une boîte à outils graphique abstraite, permet l'indépendance vis-à-vis des boîtes à outils graphiques du niveau du Composant Physique. Par exemple, la boîte à outils graphique AWT (Abstract Window Toolkit) écrite en Java relève de ce niveau : elle est indépendante de la boîte à outils graphique sous-jacente et un même programme peut utiliser insensiblement la boîte à outils graphique X/Motif sous Unix ou la boîte à outils graphique MacToolbox sous MacOS.
- Le Contrôleur de Dialogue est la pierre angulaire de cette architecture puisque ce composant a la charge de gérer le dialogue, c'est-à-dire l'enchaînement des tâches. Ce composant manipule à la fois les objets conceptuels et les objets de présentation nécessaires à l'interaction. Le Contrôleur de Dialogue associe un ou plusieurs objets de présentation avec un ou plusieurs objets conceptuels correspondants et réciproquement.

Ce modèle fournit une décomposition fonctionnelle canonique à gros grain. En effet, le modèle Arch structure un système interactif selon cinq niveaux d'abstraction, distinguant ceux qui relèvent du domaine de l'application de ceux qui gèrent l'interface utilisateur. Néanmoins, ce modèle n'apporte aucune précision sur la structure de ses composants comme celle du Noyau fonctionnel et du Contrôleur du Dialogue.

Le modèle PAC reprend les cinq niveaux fonctionnels du modèle Arch et structure le Contrôleur de Dialogue avec une hiérarchie d'agents PAC comme le montre la figure suivante.

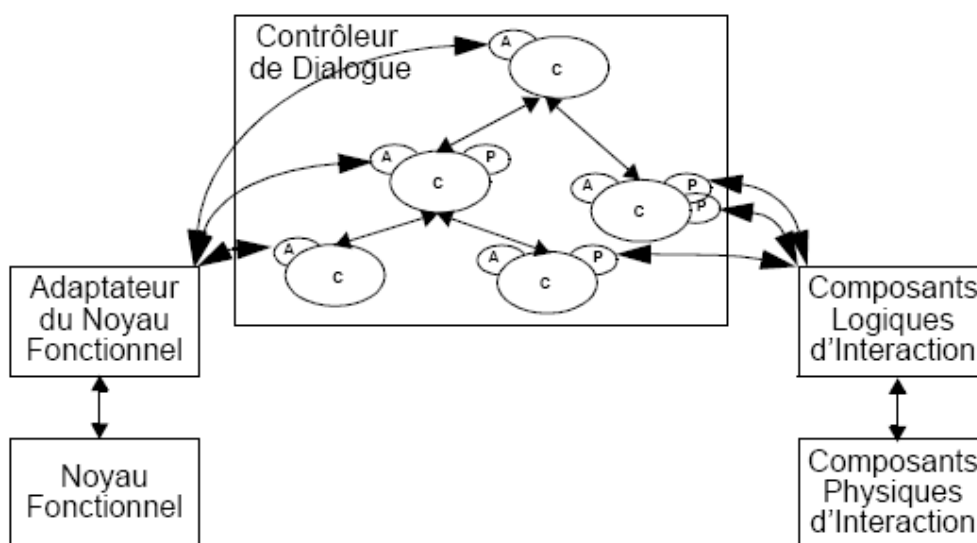


Figure 32 : Modèle d'architecture PAC (Laurillau, 2002)

Dans le modèle PAC*, le découpage fonctionnel d'un agent PAC est affiné selon les trois espaces fonctionnels du modèle du trèfle : production, communication et coordination. Ces trois facettes d'un agent PAC sont décomposées en trois parties dédiées chacune à une dimension du modèle du trèfle (Figure 33). Par exemple, la facette Abstraction est composée de trois sous-facettes dédiées respectivement à la production, à la communication et à la coordination. Cet agent est ainsi composé de trois tranches.

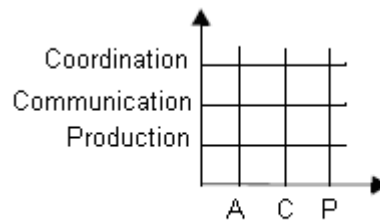


Figure 33 : Démentions fonctionnelle d'un Agent PAC (Laurillau, 2002)

Le Modèle d'Architecture Clover

Le modèle Clover résulte de la combinaison de l'approche en couches répliquées et partagées de Dewan avec la décomposition fonctionnelle du trèfle des collecticiels (Clover). Le trèfle des collecticiels définit trois classes de fonctionnalités, intitulées production, communication et coordination. Ces trois classes sont présentes dans chaque couche fonctionnelle du modèle.

Le modèle d'architecture Clover est une instance du métamodèle de Dewan par l'application des cinq couches préconisées par le modèle Arch. Le modèle d'architecture Clover compte six couches fonctionnelles : le Noyau Fonctionnel du modèle Arch est divisé en deux couches : le Trèfle Fonctionnel répliqué et privé, et le Trèfle Fonctionnel partagé et publique comme le montre la Figure 34.

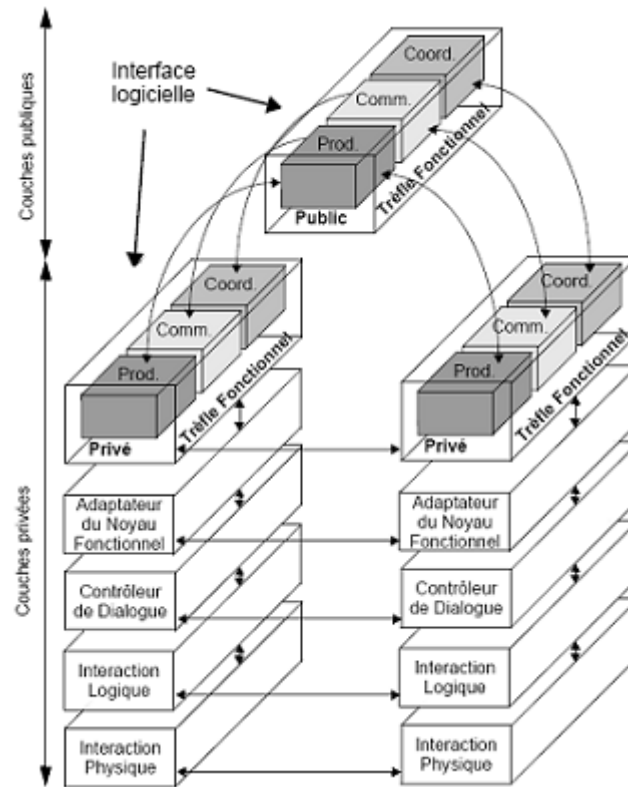


Figure 34 : Modèle d'Architecture Clover (Laurillau, 2002)

Tous les composants du modèle Arch sont répliqués à l'exception du Trèfle Fonctionnel partagé. Le Trèfle Fonctionnel partagé permet à tous les utilisateurs de manipuler les objets du domaine et d'accéder aux différents services au cours de l'interaction. A l'opposé, le Trèfle Fonctionnel répliqué gère un ensemble d'objets du domaine et l'état du système qui sont privés à un utilisateur.

L'originalité du modèle Clover réside dans la décomposition selon les facettes du trèfle des collecticiels des composants fonctionnels : Trèfle Fonctionnel partagé et Trèfle Fonctionnel répliqué. Chaque composant Trèfle Fonctionnel est composé de trois sous-composants, chacun dédié à une facette du trèfle des collecticiels. Chaque sous-composant fournit des services et gère des objets spécifiques à la production, à la communication ou à la coordination. Ces trois sous-composants sont englobés par une interface logicielle, similaire à une approche par composants comme CORBA. Le rôle de cette interface logicielle est triple :

- o L'interface logicielle englobe les trois sous-composants. Son rôle est de masquer la structuration interne du composant selon les facettes du trèfle. Cette interface permet donc la communication avec des couches voisines qui ne seraient pas structurées selon les facettes du trèfle. Ainsi, à la Figure 34, grâce à l'interface logicielle, le Trèfle

Fonctionnel répliqué peut communiquer avec l'Adaptateur de Noyau Fonctionnel qui n'est pas structuré selon les facettes du trèfle.

- L'interface logicielle gère les objets communs aux trois sous-composants. Ainsi les sous-composants Production, Communication et Coordination peuvent utiliser des objets en commun et leur appliquer des traitements qui leur sont propres.
- L'interface logicielle définit aussi un ensemble de fonctionnalités incluant les services systèmes ainsi que les fonctionnalités relevant de l'activité mono-utilisateur. En effet, ces fonctionnalités ne sont pas intrinsèquement collaboratives et ne relèvent donc pas de l'une des trois facettes du modèle du trèfle.

La communication entre couches d'une branche ou entre couches de différentes branches suit les mêmes règles prescrites par le modèle de Dewan. Cependant, les événements d'interaction et de collaboration sont différenciés selon les trois facettes : production, communication et coordination. Cet affinement permet une communication directe entre composants spécialisés. Le modèle maintient néanmoins un événement générique qui assure la communication avec les couches qui ne seraient pas structurées selon les trois facettes.

Le Méta-Modèle d'Architecture Clover

Comme le métamodèle de Dewan, le métamodèle Clover structure un collecticiel en un nombre variable de couches (Figure 35), la couche L désigne le Trèfle Fonctionnel répliqué tandis que les couches L+1 à N correspondent au Trèfle Fonctionnel partagé.

L'originalité du métamodèle réside dans la structuration des unités (Trèfle Fonctionnel) logicielles constitutives de la racine et des branches. Un Trèfle Fonctionnel est décomposé en trois sous-composants (production, communication et coordination) englobés par une interface logicielle. La communication entre ces unités est réalisée grâce à des événements qui sont soit génériques, soit dédiés à la production, à la communication ou à la coordination.

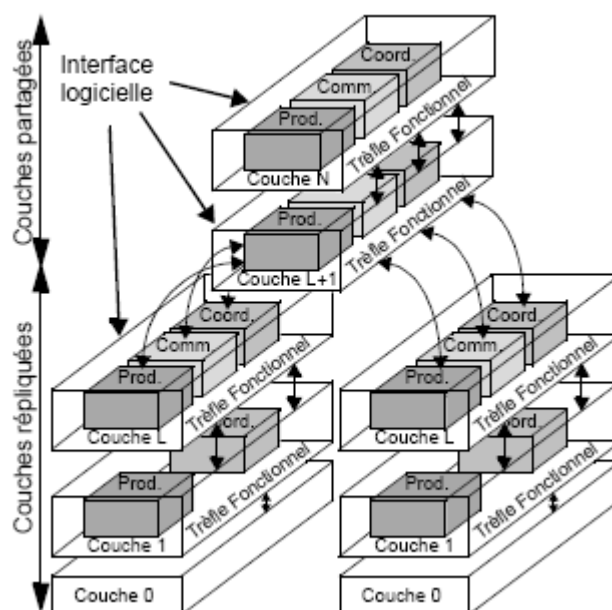


Figure 35 : Le Méta-Modèle d'Architecture Clover (Laurillau, 2002)

Le rôle principal de l'interface logicielle d'un composant Trèfle Fonctionnel est de masquer la structuration interne du composant. Ainsi un composant Trèfle Fonctionnel peut être empilé avec des composants non structurés selon les facettes du trèfle des collecticiels. En particulier, les couches les plus basses dépendantes du matériel ne sont généralement pas dédiées au développement de collecticiels et donc ne sont pas structurées selon les trois facettes du trèfle des collecticiels.

En synthèse, l'unité logicielle constitutive du métamodèle, un Trèfle Fonctionnel, contient une interface logicielle qui à son tour englobe éventuellement un, deux ou trois sous-composants, respectivement dédiés à la production, communication et coordination. La cohabitation de styles offre l'avantage de choisir le style le mieux adapté à telle ou telle couche de l'organisation structurelle. Inversement, l'hétérogénéité implique de maintenir une interface logicielle au niveau de chaque couche.

8.3 Génération d'un modèle d'architecture

Pour élaborer un modèle d'architecture logicielle d'un collecticiel de simulation, nous avons adopté une approche qui nous semble optimiste. Cette approche consiste à étudier la possibilité d'utilisation d'un modèle existant adéquat à notre problème parmi ces modèles, puis l'adapter si nécessaire.

Dans la section précédente nous avons présenté les modèles d'architecture logicielles pour les collecticiels. Parmi ces modèles nous avons présenté deux méta-modèles : Le méta-modèle de Dewan et le méta-modèle d'architecture Clover. Le deuxième est une extension du premier. Pour générer un modèle d'architecture logicielle de collecticiel de simulation, nous avons utilisé le méta-modèle d'architecture Clover.

Architecture logicielle globale

Partant des hypothèses définis au début de ce chapitre, notre futur collecticiel doit être à base de web. Cette dernière technologie repose sur la technologie Client/serveur : un serveur fournit de l'information à ses clients, avec éventuellement certaines restrictions d'accès. Pour pouvoir être utilisé sur toutes les plates-formes (PC, Macintosh, stations de travail UNIX, etc.), et ainsi éviter des problèmes de portabilité, l'information est interprétée localement par une machine virtuelle du poste client.

Pour élaborer générer l'architecture logicielle de notre futur collecticiel, nous avons appliqué le méta-modèle d'architecture Clover, présenté précédemment. L'architecture résultante est composée des Trèfles Fonctionnels publics et privés. La Figure 36 présente l'architecture logicielle de notre futur collecticiel. Le Trèfle Fonctionnel public est centralisé sur le serveur et le Trèfle Fonctionnel privé est décomposé en deux parties, une au niveau du serveur et l'autre au niveau du client.

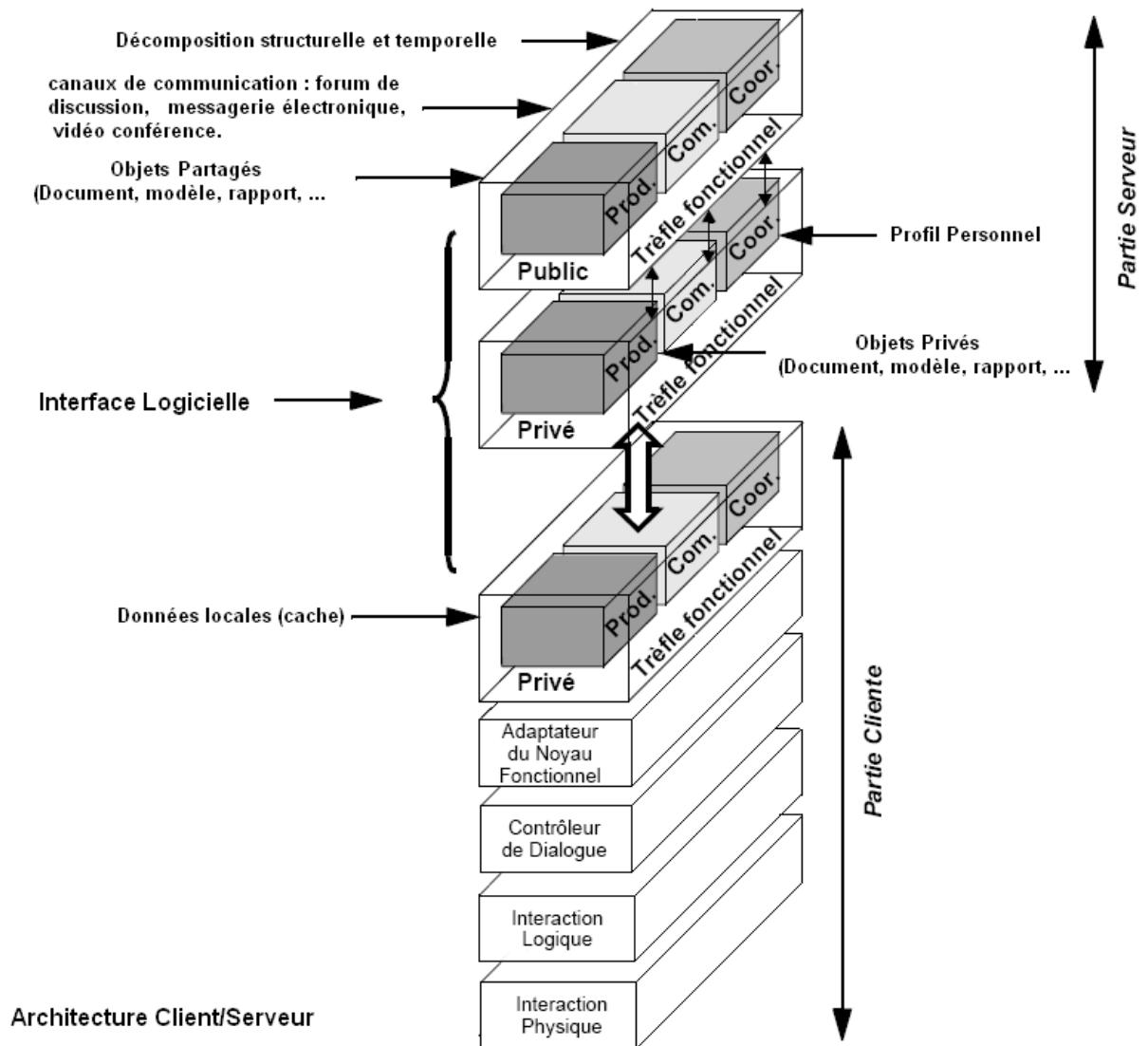


Figure 36 : Modèle Architecture logicielle pour un collecticiel de simulation

Nous présentons notre architecture en partant de la couche la plus haute vers la couche la plus basse. Au plus haut niveau d'abstraction, le Trèfle Fonctionnel public, est composé d'un noyau fonctionnel dédié à la production, d'un noyau fonctionnel dédié à la communication, d'un noyau fonctionnel dédié à la coordination et d'une interface logicielle.

- **Composant public dédié à la production** : ce composant maintient l'ensemble des objets partagés. Les principaux concepts manipulés sont les Modèles, Résultats, les documents, Rapports, Agenda de projet etc. De nombreuses fonctions sont disponibles pour manipulé (voir Tableau 7).

- **Composant public dédié à la communication** : ce composant gère différents canaux de communication. Le canal de communication est l'unique concept manipulé par ce composant. Un client peut s'abonner ou se désabonner à un canal de communication et peut y poster des messages. Le futur collecticiel offre plusieurs canaux de communication : forum de discussion, messagerie électronique et vidéo conférence.
- **Composant public dédié à la coordination** : Ce composant a la responsabilité de coordonner les membres et les groupes d'un projet de simulation. Il dispose de mécanismes gérant les accès concurrents aux données partagées et notifie tous les clients des changements d'état. Les principaux concepts manipulés par ce composant sont les concepts suivant : Planning prévisionnel & suivi de l'avancement du projet (décomposition structurelle et temporelle, Répartition des ressources), phase ou étape de travail, Rôles et compétences, etc. ... Le futur collecticiel doit offrir les fonctionnalités qui permettent de manipuler ces concepts (Voir Tableau 6).

Le Trèfle Fonctionnel public est empilé au-dessus du Trèfle Fonctionnel privé. Ce dernier est divisé en deux parties, l'une au niveau client et l'autre au niveau du serveur. Le Trèfle Fonctionnel privé localisé sur le client maintient un état local à l'application et une copie de l'état privé située sur le serveur. Un mécanisme de cache permet d'optimiser les performances à l'exécution. Le Trèfle Fonctionnel privé situé sur le serveur est constitué de trois sous-composants, dédiés à la production, la communication et à la coordination. Ces composants maintiennent les données privées d'un utilisateur.

Les autres couches : l'Adaptateur du Noyau Fonctionnel, le Contrôleur de Dialogue, Interaction Logique et Interaction Physique, ne sont pas décomposées selon les trois facettes du trèfle des collecticiels.

Architecture de la partie cliente

La partie cliente est composée du Trèfle Fonctionnel privé, de l'Adaptateur du Noyau Fonctionnel, Contrôleur de Dialogue, couche de l'interaction logique et la couche d'interaction physique. Les deux couches les plus bas ne jouent pas un rôle essentiel dans ce contexte. Comme l'architecture du modèle clover est orienté agent, l'architecture conceptuelle de la partie cliente est constituée d'une hiérarchie d'agents comme le montre la figure suivante :

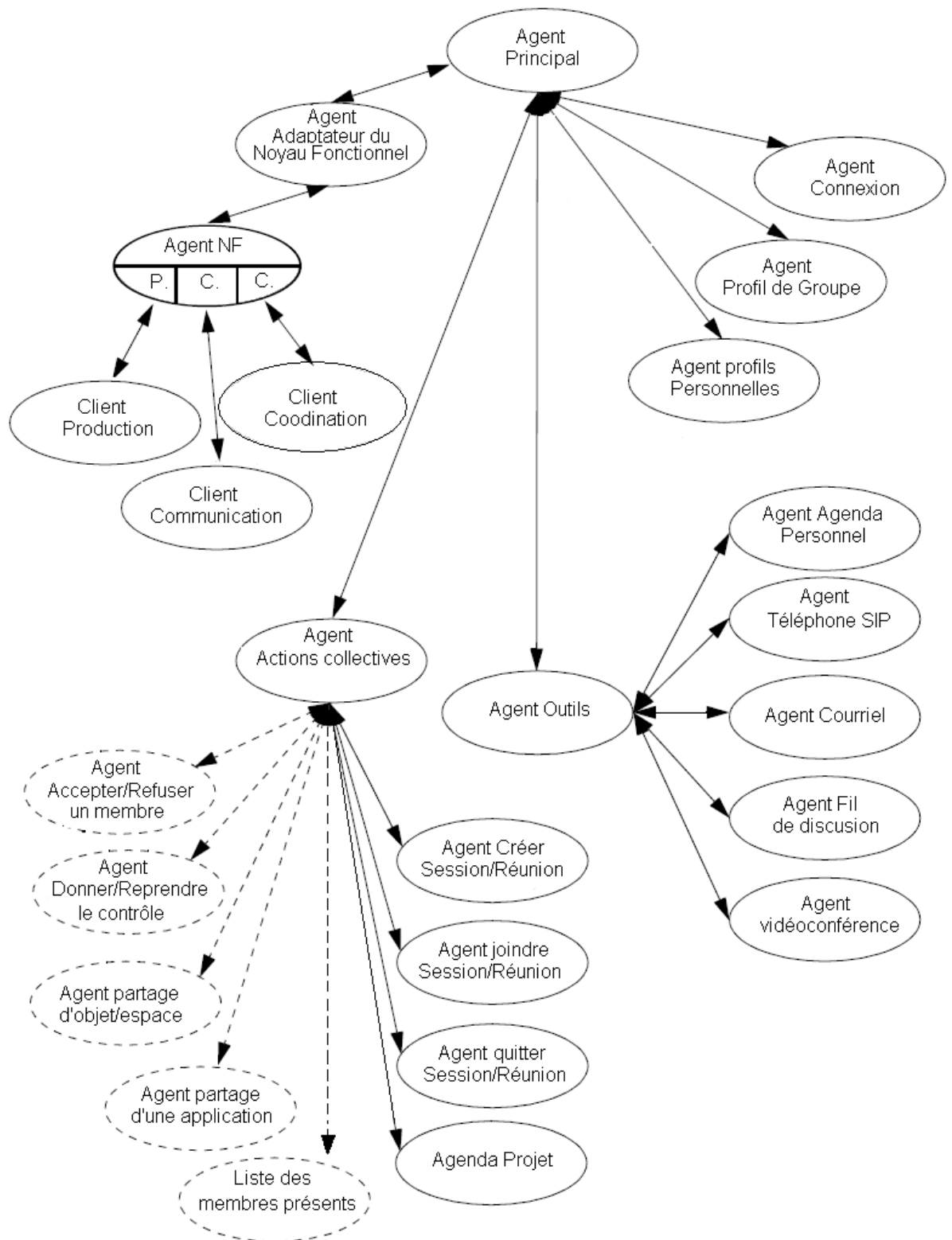


Figure 37 : Modèle de hiérarchie d'agents constituant la partie cliente

- **Adaptateur du Noyau Fonctionnel** : L'Adaptateur du Noyau Fonctionnel joue un rôle de médiateur entre le Noyau Fonctionnel et le Contrôleur de Dialogue. Les données échangées avec le Noyau Fonctionnel sont les objets du domaine que le Noyau Fonctionnel exporte vers l'utilisateur. Les données échangées avec le Contrôleur de Dialogue sont des objets conceptuels correspondant à une représentation mentale de l'utilisateur des objets du domaine. Le rôle de l'agent Adaptateur du Noyau Fonctionnel dans notre architecture est chargé de faire la correspondance entre le concept d'utilisateur employé par le Noyau Fonctionnel et sa représentation dans le composant de Contrôleur de Dialogue. Il est de préférence de donner la possibilité aux utilisateurs de personnaliser cette présentation.

- **Contrôleur de Dialogue** : Le Contrôleur de Dialogue est constitué d'une hiérarchie d'agents organisée en trois catégories : les agents dédiés aux actions collectives, les agents dédiés aux outils et les autres agents : deux agents pour la gestion des profils personnelles et du groupe (agents profils personnelles et profil d'un groupe), et un agent Connexion pour connecter au collecticiel. Les agents dédiés à l'action collective sont gérés par un agent ciment (agent ciment : Un agent situé à un nœud de la hiérarchie n'implémentant que la facette Contrôle et/ou la facette Abstraction.) l'agent Actions collectives. Comme le montre la Figure 37, le nombre d'agents fils de l'agent évolue dynamiquement au cours de l'interaction par ajout et suppression d'agents. Le nombre d'agents varie en fonction de mode de coopération. Par exemple, lorsqu'un membre rejoint une session, de nouvelles actions sont disponibles et cela se traduit par l'apparition de nouveaux agents fils. Réciproquement, lorsqu'un membre quitte une session de travail, des agents fils sont détruits et retirés de la hiérarchie. Néanmoins, les trois agents suivants ne peuvent être supprimés : l'agent Créer une session/réunion, l'agent Joindre session/réunion et l'agent Quitter une session/réunion qui correspondent aux modes de coopération session, réunion et étroite. Les agents dédiés aux outils sont gérés par un agent ciment, l'agent Outils. Les cinq agents fils sont : l'agent agenda personnel, l'agent téléphone SIP, l'agent courriel, l'agent fil de discussion et l'agent vidéo de conférence.

9 Spécifications externes et propriétés ergonomiques

Les spécifications externes décrivent le système tel qu'il sera perçu et utilisé par l'utilisateur. Les requis utilisateurs concernent l'utilisabilité du collecticiel et donc l'interface du collecticiel. Communément, l'utilisabilité se décline en propriétés ergonomiques.

9.1 Critères ergonomiques

Les critères d'ergonomie sont un raffinement d'un des critères de qualité de Génie Logiciel : l'utilisabilité. Ceux-ci sont très largement employés dans la conception des systèmes en IHM. Ces critères s'organisent autour de trois familles de facteurs : apprentissage, souplesse et robustesse.

Dans ce point, nous nous intéressons à des critères d'ergonomie spécifiques aux collecticiels. Cependant, certains critères applicables aux systèmes mono-utilisateurs doivent être aussi vérifiés.

L'Observabilité publiée : Dans une activité de groupe, l'observabilité se heurte au concept de protection de l'espace privé. L'espace privé se définit comme l'ensemble des variables d'état "personnelles" (par exemple, le fait qu'un membre est dans son bureau en train de lire son courrier électronique). Le caractère personnel d'une variable se définit dès l'analyse du problème. A l'évidence, l'observabilité des variables personnelles peut être pertinente pour le groupe mais potentiellement contraire au principe du respect de l'intimité. Aussi convient-il d'introduire la notion d'observabilité publiée : les variables d'état personnelles pertinentes pour autrui sont rendues observables seulement si le propriétaire en autorise la publication. L'autorisation de publication peut être statique ou dynamique. Dans notre cas, ce critère est très important. Des informations observables de l'utilisateur, nous considérons trois concepts importants :

- Observable : ce qui doit être perçu par tout utilisateur,
- Publiable : ce qu'un utilisateur décide de rendre observable de lui même,
- Filtrage : la forme de ce qui est observable. Les informations ainsi filtrées donnent de l'utilisateur l'apparence qu'il a choisie.

La différence entre un filtrage et une information publiée se situe dans la véracité des informations : quand on publie une information, celle-ci est a priori exacte, quand on la filtre, on la modifie, on la modèle à sa propre façon.

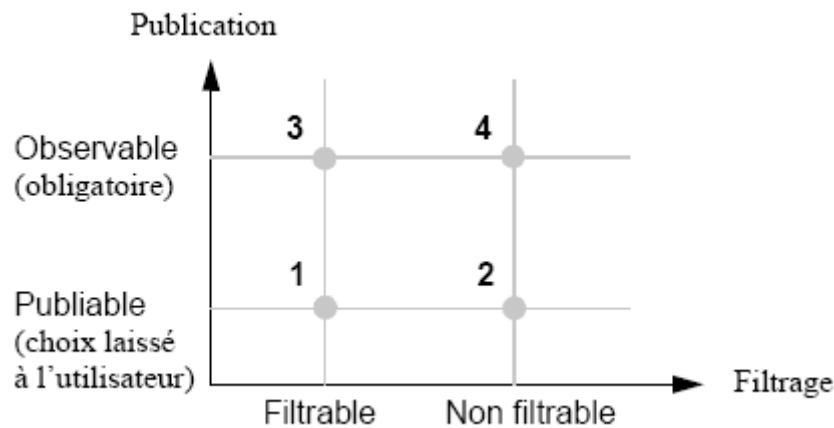


Figure 38 : Quatre niveaux d'observabilité (Laurillau, 2002).

L'observabilité peut être définie par quatre niveaux (Laurillau, 2002) comme présente la Figure 18 : les quatre niveaux sont définis suivants 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.

Nous distinguons les informations concernant les membres d'un projet de simulation :

- Informations caractérisant un membre :
 - Une forme (représentation visuelle) : avatar, photographie, etc.,
 - Des informations d'identification : ensemble des données propres au membre telles que le nom & prénom, l'adresse, l'e-mail, qualification scientifique et professionnelle, etc.,
 - Des informations concernant son statut dans le projet : responsabilités, rôles, etc.,
 - Le(s) groupe(s) auquel(s) il appartient.
 - Liste des travaux réalisés dans le cadre d'un projet

- Informations caractérisant un groupe :
 - Nom du Groupe,
 - Responsable du groupe.
 - Liste des membres,
 - Statut dans le projet : spécialité, mission, etc.,
 - Liste des travaux réalisés dans le cadre d'un projet.

	Seul	Associe a groupe
Photographie	1	1
Nom de l'Utilisateur	4	4
Travaux (personnel)	1	1
Travaux (groupe)	N/A	4
Informations (personnel)	3	3
Informations (groupe)	N/A	4
Statut (personnel)	N/A	4
Statut (groupe)	N/A	4

Tableau 8 : Matrice des concepts observables.

Suivant les quatre niveaux d'observabilité, nous détaillons, dans le Tableau 5, les informations publiables et filtrables pour un membre et pour les groupes auxquels il appartient, en fonction de son appartenance à un groupe et en fonction du rôle au sein du groupe. L'affectation de la valeur 4 dans le tableau traduit que toutes ces informations doivent être observables car elles assurent la validité de la propriété d'*awareness*, quelque soit le statut du membre ou du groupe. Quelque soit son statut, l'utilisateur a la possibilité de contrôler totalement la publication et le filtrage des informations concernant son activité, ce qui justifie la valeur 1 pour toute la ligne "travaux personnel". En ce qui concerne les informations personnelles propres à un membre, nous jugeons qu'il est nécessaire qu'elles soient observables, mais filtrables. Ce qui se traduit par la valeur 3 pour toute la ligne "informations (personnel)".

Awareness : Dans un collecticiel, à la différence d'une application mono-utilisateur, plusieurs personnes peuvent agir en même temps sur les mêmes objets. L'utilisateur unique a naturellement conscience des actions qu'il entreprend. Par contre, dans un environnement partagé, les acteurs n'ont pas spontanément connaissance des agissements d'autrui. Dans le domaine du TCAO, on généralise cette connaissance par la notion de « sensation de présence », ou encore par la notion de « conscience ». On a conscience d'un individu, ou d'un groupe d'individus, si on connaît la position et l'activité courante de cet individu, ou de chaque membre du groupe, dans l'espace partagé. Cette notion traduit la possibilité pour chaque utilisateur d'être informé de l'état ou des actions des autres utilisateurs de façon périphérique. Cette notion est une information critique pour l'établissement du succès d'un projet de simulation. Elle permet, en effet, de réguler et de coordonner les comportements de chaque membre. Dans toutes les situations, le collecticiel future doit garantir la cohérence des données et fournir à chacun des utilisateurs une représentation de l'activité du groupe sous les trois formes suivantes :

- o Conscience de la présence des membres du groupe et leur disponibilité dans le travail coopératif,
- o Conscience des actions réalisées par le membre du groupe,
- o Conscience des effets consécutifs à ces actions.

Contrôle d'accès : Cette propriété se traduit par la possibilité de gérer 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. Dans le cas d'un projet de simulation, le chef de projet a le pouvoir de gérer les droits ou de déléguer un autre membre du projet.

WYSIWIS et couplage de l'interaction : se traduit en principe 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. Dans notre cas, le collecticiel futur doit permettre la possibilité de paramétrage de cette contrainte selon les différentes modes de coopération dans un projet de simulation.

9.2 Métaphore de l'Interface utilisateur

Les interfaces Home Machine à base de fenêtres exploitant la métaphore de bureau se sont imposés avec les environnements tels que X-Windows sur station de travail et MicroSoft Windows sur PC. Actuellement, de nombreux travaux sont en cours en vue de définir de nouvelles métaphores qui permettront de construire des IHM plus performantes orientées surtout vers les applications de groupe. La métaphore de bureau (2D) a tendance à être substituée par la métaphore de pièce ou Room (3D) pour élargir l'espace de travail et ainsi augmenter la densité de l'information. Une autre approche qui s'éloigne radicalement de l'utilisation bureautique de l'informatique est celle

utilisant la réalité virtuelle et avec laquelle l'écran est remplacé par un casque et de nouveaux outils d'interaction entre l'utilisateur et la machine sont introduits. Nous estimons que dans le cas d'un collecticiel de simulation, une IHM basée sur la métaphore de bureau qui est aujourd'hui rentrée dans les mœurs, est largement suffisante.

La prise en compte de la configuration matérielle dont dispose chaque membre permet au collecticiel d'associer une interface graphique propre à chaque membre. Par exemple, un membre n'ayant pas de matériel pour exploiter le son, ne verra pas au niveau de son interface les icônes permettant d'activer les outils de traitement du son. De même que la prise en compte des privilèges permet d'activer ou de désactiver certaines icônes de l'interface graphique correspondant aux activités associées à des tâches pour lesquelles un membre n'a pas reçu de privilèges. Par exemple, un membre qui n'a pas eu le privilège d'exécuter une activité de la tâche de construction d'un modèle de simulation, ne verra pas au niveau de son interface les icônes représentant les outils de construction d'un modèle de simulation (Editeur graphique de modèles).

9.3 L'interface et le modèle du trèfle

Le modèle du trèfle fonctionnelle est un support utile à la spécification fonctionnelle des collecticiels. Dans sa thèse de doctorat Laurillau (Laurillau, 2002) que cette décomposition fonctionnelle est difficilement applicable à la conception de l'interface homme-machine. Cette difficulté n'implique pas que nous pouvons appliquer la décomposition fonctionnelle selon les trois espaces du modèle du trèfle aux composants de l'interface de notre futur collecticiel. En effet, de nombreux collecticiels proposent des objets graphiques d'interaction combinant les trois facettes du modèle du trèfle (pour plus de détail voir (Laurillau, 2002)).

10 Conclusion

Concevoir un outil logiciel destiné à supporter le travail coopératif dans la conduite d'un projet de simulation, c'est mettre en confrontation deux univers cognitifs distincts ; l'univers de la recherche sur le TCAO et l'univers de simulation, et déterminer par conséquent ce que le premier peut apporter au second. L'apport n'est d'ailleurs pas à sens unique : La méthode de conduite d'un projet de simulation se révèle être un champ d'application pour la communauté scientifique spécialisée dans le TCAO.

En exploitant notre expertise dans les deux domaines (TCAO et simulation), nous avons essayé, dans ce chapitre, de contribuer à la résolution du problème traité dans cette thèse, et ce, en proposons un modèle de collecticiel de simulation. Ce modèle est à notre avis unique dans la mesure où aucun travail similaire n'existe. Nous estimons qu'il peut servir comme base à toute démarche d'implémentation d'un collecticiel dans le domaine de la conduite de projet de simulation. Bien évidemment un modèle n'est qu'un modèle qui peut très bien être critiqué, remis en cause ou enrichi.

Chapitre n° 5

Implémentation

V Implémentation

1 Introduction

Conventionnellement la partie expérimentale est placée à la suite des parties consacrées à la modélisation et à la création de prototype (dans le but de valider ces dernières, d'en évaluer la pertinence et la performance, et de comparer une proposition nouvelle à des propositions antérieures qu'elle est censée améliorer). Dans notre cas, et comme vous pouvez l'imaginer, la réalisation d'un collecticiel fiable et prêt à l'emploi est impossible vue le temps imparti. Sans oublier que c'est un grand travail de groupe. En revanche, nous avons utilisé un collecticiel fiable prêt à l'emploi pour proposer une implémentation et de conduire une série d'expérimentation.

2 Choix de collecticiel

Au cours de cette recherche sur le choix d'un collecticiel, nous avons été amenés à définir une grille d'analyse fonctionnelle d'un collecticiel pour argumenter notre choix d'un collecticiel permettant de proposer une implémentation d'un collecticiel simulation. Pour ce faire, nous avons dressé une liste des fonctionnalités qui nous semblaient pertinentes au regard de notre connaissance de la situation à supporter. Cette grille est largement inspirée de l'analyse menée par Olivier Malcurat (Malcurat 2001) sur un sujet que nous estimons similaire à la notre.

2.1 Grille d'analyse d'un collecticiel

Notre grille comporte cinq rubriques : les données partagées, les services de communications, les services de coordination, la conscience de groupe et l'architecture du système. Chaque rubrique est composée de fonctionnalités plus ou moins supportées par le collecticiel.

Les données partagées

Dans cette rubrique, nous trouvons les fonctionnalités suivantes : les objets et relations, le contrôle de concurrence (des objets), le mécanisme de versions, le contrôle d'accès :

- o Les objets et relations. Le collecticiel met des types (ou des classes) d'objets à disposition des utilisateurs. Dans un lecteur de courrier électronique, ce sont les messages, les entêtes de messages, les fichiers attachés, le carnet d'adresses, etc. Dans un collecticiel de

projets partagés, ce sont les répertoires, les fichiers, les auteurs, les discussions structurées, etc. Tous ces objets maintiennent des relations entre eux (ex. un répertoire contient des fichiers et des répertoires). La complexité des objets et la pertinence des relations participent de la richesse fonctionnelle du collecticiel. Les annotations (ou méta-données) par exemple sont un type particulier d'objets qui s'attachent à des objets pour les qualifier, les commenter, les critiquer.

- Le contrôle de concurrence. L'exécution simultanée d'un collecticiel par plusieurs utilisateurs pose le problème de l'accès concurrent des données. Si le même fichier peut être modifié en même temps par deux utilisateurs distants, comment garantir un résultat cohérent et conforme à ce que souhaitent les deux utilisateurs ? Les stratégies développées pour répondre à ce problème sont multiples ; toutes ne sont pas également puissantes, ni aussi complexes. La plus simple consiste à bloquer l'accès d'un objet dès lors qu'il est utilisé par un premier utilisateur. D'autres stratégies permettent de s'affranchir de cette contrainte ; la stratégie du « copier-modifier-unifier » (copy-modify-merge) permet à un même objet d'être développé simultanément et isolément par plusieurs utilisateurs, à condition toutefois que le dernier « accesseur » de l'objet fasse l'opération d'unification des différentes versions.
- Le mécanisme de versions. Certains objets ont pour vocation d'être transformés et augmentés ; c'est le cas des objets de type fichiers. Le mécanisme de versions permet de conserver la trace de l'ensemble des versions produites (parfois de manière concurrente) et le lien temporel qui les unit. Certains collecticiels ne gèrent pas les versions ; d'autres la gèrent de manière linéaire comme une suite simple de versions, d'autres de manière arborescente avec « branches » multiples, des jonctions, etc.
- Le contrôle d'accès. Il est souvent nécessaire de restreindre l'accès aux objets partagés pour des raisons évidentes de sécurité. De nombreux collecticiels sécurisent leurs accès par la procédure de l'identification (« login » et mot de passe) ; l'utilisateur préalablement enregistré accède selon les droits qu'il a reçu à des objets sélectionnés. Ces droits d'accès déterminent des actions autorisées (voir, lire, modifier, effacer, créer, etc.) sur les objets. La politique de droits est parfois plus complexe ; un même utilisateur peut avoir des droits différenciés sur les objets et par exemple, avoir un droit de lecture seul sur un objet et un droit de modification sur un autre.

Les services de communication

L'informatique communicante profite de l'informatique multimédia ; selon le type de collecticiel, les utilisateurs échangent des textes, des images fixes, des vidéos, des sons, ou tout autre type de

données (ex. fichiers). L'échange des données est soit synchrone (co-présence de l'émetteur et du récepteur), soit asynchrone (absence du récepteur au moment de l'émission). Le croisement des critères de type de média et de synchronisation résulte en un tableau des types de services de communications possibles.

	Synchrone	Asynchrone
Texte	dialogue textuel (chat)	courrier électronique
Vidéo	Vidéo webcam, flux de données (streaming)	fichiers attachés aux courriers électroniques
Image	tableau blanc	fichiers attachés aux courriers électroniques
Son	conversation téléphonique	courrier vocal (voice mail)
Autres données	tableau blanc, partage d'applications	fichiers, plannings

Tableau 9 : Services de communication (Malcurat 2001).

Actuellement, peu de collecticiels supportent plusieurs services. La complexité technologique des solutions existantes amène les concepteurs de ces collecticiels à concentrer leurs efforts sur une seule technologie.

Gestion de la coordination

Les services de coordination ont pour but de permettre aux utilisateurs, tout au moins à certains d'entre eux, de prescrire aux autres des tâches. Il existe plusieurs niveaux de gestion de la coordination. Nous en distinguerons trois :

- Le niveau manuel. C'est le cas de l'agenda partagé sous sa forme la plus simple. Les tâches ou événements sont réparties à des dates précises ; charge au personnel de vérifier s'il est concerné ou non. Les tâches prescrites peuvent être très librement formalisées dans la mesure où elles ne seront lues et comprises que par des « agents » humains.
- Le niveau semi-automatique. Les services semi-automatiques permettent de formuler librement les tâches ; le système se charge des procédures de notification ou de rappel de tâches. C'est le cas de certains calendriers électroniques évolués ou des systèmes de « Post-It » partagés (To Do List).
- Le niveau automatique. Les tâches sont complètement formalisées ; elles deviennent alors « compréhensibles » par le système qui peut alors prendre le contrôle de l'exécution d'un ensemble structuré de tâches. Les systèmes les plus aboutis dans ce domaine sont les gestionnaires de flux de tâches (workflow systems) dans lesquels on définit des procédures (ensemble de tâches et des conditions d'exécution de tâches) antérieurement

à leur exécution. Leur efficacité est maximale dans des situations où toutes les éventualités sont envisagées (ex. procédure d'achat sur Internet).

La conscience de groupe

Par définitions de la conscience de groupe est « la compréhension des activités des autres, qui fournit un contexte pour sa propre activité ». Elle permet d'encourager la communication spontanée et informelle entre les personnes et les renseigne en permanence sur les événements importants afin de prendre les meilleures décisions. Cette notion étant très générale, il est nécessaire d'en préciser le contenu et, à la suite de Malcurat (Malcurat 2001), nous distinguons quatre sous-types de conscience de groupe :

- La conscience informelle de groupe : connaissance élémentaire des personnes qui nous entourent physiquement ou virtuellement, de l'endroit où elles se trouvent.
- La conscience de la structure de groupe : connaissance de la position de chacun dans l'organisation virtuelle, de leurs rôles et responsabilités, des processus qui ont cours habituellement, etc.
- La conscience sociale de groupe : information qu'une personne a sur d'autres personnes avec qui elle entretient une relation sociale, comme une conversation : leur humeur, l'intérêt qu'elle porte à une chose, leur attention.
- La conscience de l'espace de travail : connaissance détaillée et mise à jour des actions entreprises par d'autres. Dans un collecticiel synchrone comme un tableau blanc partagé, il s'agit par exemple de savoir qui est entrain de dessiner, qui entre dans une session ou en sort, etc. Dans un collecticiel asynchrone comme les espaces de projets persistants (persistent shared workspace), il peut s'agir de savoir qui a modifié tel objet et quand.

L'architecture système

L'architecture d'un système dans ses aspects les plus élémentaires, parce qu'elle détermine indirectement certaines fonctionnalités du collecticiel, nous importe ici. Nous identifions trois critères qui s'avèrent souvent décisifs dans le choix d'un collecticiel :

- Environnements d'exécution du client. A notre connaissance, les systèmes informatiques sont dominés par deux principaux environnements (Windows et MacOS), les autres étant très largement marginaux (ex. Unix, Linux). Évidemment, les collecticiels qui ne s'exécutent que dans un seul de ces environnements excluent indirectement les utilisateurs d'autres environnements.

- Clients. Le client est le programme d'interface exécuté par l'utilisateur du collecticiel. Les clients web (webbased groupware) s'exécutent depuis un navigateur Internet ; l'avantage de cette formule est son universalité, toutes les personnes ayant accès à Internet ont un navigateur Internet et sa simplicité d'emploi ; l'inconvénient réside dans la faiblesse des interfaces graphiques et la difficulté à supporter la communication synchrone. Pour remédier à ces problèmes, certains clients « web » sont écrits, totalement ou en partie, dans le langage Java. Le programme client peut être écrit spécifiquement pour l'environnement dans lequel ils s'exécutent ; contrairement aux deux solutions précédentes, ils nécessitent l'installation, parfois le téléchargement, d'un programme et des mises à jour occasionnelles. Ses avantages sont ceux que l'on peut déjà obtenir avec les clients « java » avec en plus une meilleure intégration à l'environnement d'exécution (ex. possibilité de couplage avec des logiciels installés, interface graphique standard, etc.).
- Serveurs. Le serveur est le programme qui dialogue avec les programmes clients ; il est le cœur du collecticiel car il centralise et pérennise les données (référentiel unique) de tous les utilisateurs. Dans la grande majorité des collecticiels actuels, le serveur est hébergé par une société qui « loue » (ou « prête » pour des évaluations) ses services aux utilisateurs. Dans ce cas, ces derniers ignorent la localisation du serveur et n'ont aucun contrôle sur la confidentialité des données qu'ils y entreposent. Quelquefois, le programme serveur est mis à disposition des utilisateurs pour une installation sur une machine serveur de leur choix ; par rapport à l'offre précédente, cela offre l'avantage de ne pas avoir à louer un service et d'avoir une capacité de stockage très importante ; les inconvénients sont l'obligation de disposer d'une machine serveur avec une connexion permanente.

L'autre architecture, en pleine émergence aujourd'hui, prend le nom de « poste à poste » (Peer-to-Peer) sur lequel nous ne nous étendrons pas. Cette architecture logicielle s'affranchit d'une centralisation des données par un ou plusieurs serveurs. Les postes clients établissent des connexions directes entre eux. Si les problèmes de confidentialité des données et de dépendance vis-à-vis d'une société de services y trouvent un début de solution, la pérennité des données n'est assurée que si les utilisateurs, à la fois clients et serveurs, maintiennent une connexion permanente au réseau (ex. connexion ADSL ou câble).

2.2 Le collecticiel retenu

Après avoir analysé et comparé plusieurs collecticiels disponibles sur le marché (cf. Annexe 1 : grille d'analyse des collecticiels), notre choix de collecticiel d'implémentation s'est porté sur le

système BSCW (Basic Support Cooperative Work). Ce choix est motivé d'une part par l'incertitude quant aux moyens informatiques de tous les participants du projet, ainsi que la simplicité, l'universalité de son accès et sa disponibilité. Cette solution dispense en effet les membres de l'installation d'un logiciel client spécifique sur les machines dont ils désirent se servir. Elle leur permet également de travailler depuis des endroits différents.

Le tableau suivant résume l'application de cette grille sur le système BSCW.

1 : Données partagées	
Objets et dépendances	Organisés hiérarchiquement en répertoires. Fichier (métadonnées : description textuelle, notation, format), fil de discussion libre ou attaché à un fichier particulier, membres authentifiés, date de réunion (date de début et date de fin, lieu de la réunion, forme synchrone ou asynchrone), URL
Contrôle de concurrence	Possibilité facultative de bloquer les modifications sur un fichier (locking)
Versions	Possibilité de « versionner » les fichiers (versionnement linéaire et arborescent) ; gestion automatique et débrayable des indices de versions ; possibilité d'attacher un commentaire textuel à chaque nouvelle version
Contrôle d'accès	Authentification par mot de passe. Droits d'accès sur les répertoires (notions de membres associés à un répertoire) et possibilité de raffiner l'accès sur un fichier particulier. Définition des niveaux d'accès (correspondant à une liste d'actions permises)
2 : Services de communication	
Synchrone	uniquement par adjonction logicielle (extension java)
Asynchrone	directement par fils de discussion ; indirectement par les description de fichiers et les commentaires de versions
3 : Gestion de la coordination	
Manuelle	Par fils de discussion
Semi-automatique	Envoi des dates de réunion par courrier électronique, réponse sur le serveur (présent ou absent)
Automatique	Possibilité de s'abonner aux rapports journaliers des changements intervenus (envoi d'un courrier électronique)
4 : Conscience de groupe	
Conscience de l'espace de travail	Tout changement dans la base est notifié (icône « new » à côté des répertoires et des fichiers modifiés). Cette gestion du changement est sélective et s'opère en fonction de chaque utilisateur (observateur), c'est-à-dire que ce qui est nouveau pour un ne l'est plus forcément pour les autres s'ils ont déjà pris en compte ce changement
5 : Architecture du système	
Environnement d'exécution	Tous.
Client	Navigateur Internet (interface html)
Serveur	BSCW n'est pas à proprement parler un service (adresse unique) ; le serveur est disponible (gratuitement pour l'éducation) et doit être installé sur

	un poste serveur
6. Besoins spécifiques au projet de simulation	
Gestion hors-connexion	Obligation de se connecter au serveur
Partage d'application	Non prévue ; l'utilisation d'un outil externe (MS NetMeeting) peut combler ce manque dans une certaine mesure
Rapports avec des intervenants extérieurs	Possibilité de créer des répertoires publics (accès à des personnes anonymes) ; invitation des personnes non enregistrées par courrier électronique

Tableau 10 : Fiche d'évaluation du collecticiel BSCW.

3 Le BSCW

Au cours de ces dernières années, la recherche dans le domaine du TCAO a permis une meilleure compréhension des problèmes de la coopération électronique au sein des groupes dans divers environnements. Les études dans ce domaine ont montré l'importance des espaces d'information communs (souvent appelés les espaces de travail partagés) en particulier dans les groupes géographiquement distribués et moins organisés. Les groupes utilisent de tels espaces de travail pour la collection et la structuration de tout genre d'information qu'ils doivent réaliser dans le but de leur collaboration (par exemple, des documents, des graphiques, des bilans, des tables, ou logiciel).

Ces espaces de travail supportent principalement les modes de communication asynchrones. Ce mode est normalement le plus important pour la coopération entre les chercheurs puisque dans un tel environnement, une coopération consiste souvent à réaliser des activités parallèles et faiblement couplées des différents membres de groupe. Les types de coopération synchrones tels que les vidéoconférences ou les réunions électronique sont habituellement d'importance moindre mais devraient également être soutenues dans une certaine mesure.

Le BSCW ou (Basic Support for Cooperative Work), Développé par les laboratoires de l'allemand GMD, est un collecticiel qui fait office de repère dans la recherche sur le TCAO. Ce collecticiel est basé sur la métaphore des espaces de travail partagés.

Bien que le système BSCW supporte principalement les modes de communication asynchrones, il offre également quelques dispositifs pour la collaboration synchrone telle que les informations sur la présence concourante d'autres utilisateurs aussi bien que des interfaces d'outils de communication synchrones tels que les vidéoconférences ou les réunions électronique.

Dans ce système les utilisateurs accèdent à leurs espaces de travail avec leurs navigateurs de WEB ordinaire ; l'installation du logiciel additionnel aux postes des utilisateurs n'est pas nécessaire. L'accès à l'espace partagé est calqué sur le principe de gestion des systèmes UNIX : un utilisateur est identifié par un mot de passe, un identifiant et un groupe auquel il appartient (en général, son groupe de travail). La vue sur l'espace partagé dépend du groupe dont il est membre, sachant que tous les membres ont la même vue.

3.1 Mise en œuvre du système BSCW

Le système BSCW est installé sur un serveur de Web standard : Pour la mise en œuvre des fonctionnalités du noyau du système BSCW, l'interface CGI et l'interface API des serveurs Web standard sont modifiés. De ce fait le système BSCW représente une extension des serveurs Web afin de permettre les fonctions d'un groupware. Le système est programmé entièrement par le langage interprété Python et le seul logiciel additionnel exigé pour utiliser le système en plus du serveur Web est le compilateur du langage Python.

Puisque Python offre un très bon support de modularité, l'implémentation de la fonctionnalité de noyau et l'interface utilisateur sont largement séparées, c.-à-d., sans modifier le code, l'interface utilisateur peut être adaptée largement aux besoins du client, même par des personnes qui n'ont pas des connaissances détaillées sur le code. La définition d'interface comporte un ensemble de pages HTML qui peuvent être éditées facilement. GMD fournit ces pages d'interface en Allemand et en Anglais, mais les utilisateurs du système les ont traduites pour offrir des interfaces dans d'autres langues (par exemple, français, Italien, Espagnols, Finlandais, Russes).

La conception modulaire du système permet également l'extension du BSCW de différentes manières. De nouveaux opérateurs peuvent être ajoutés pour offrir une nouvelle fonctionnalité ou pour agir en tant qu'interfaces «wrappers» à une application existante. Il est également possible de stocker de nouveaux types d'objets d'une manière claire sans modifier les routines de stockage du système. En particulier, le choix du langage Python comme langage d'implémentation permet le prototypage rapide et direct.

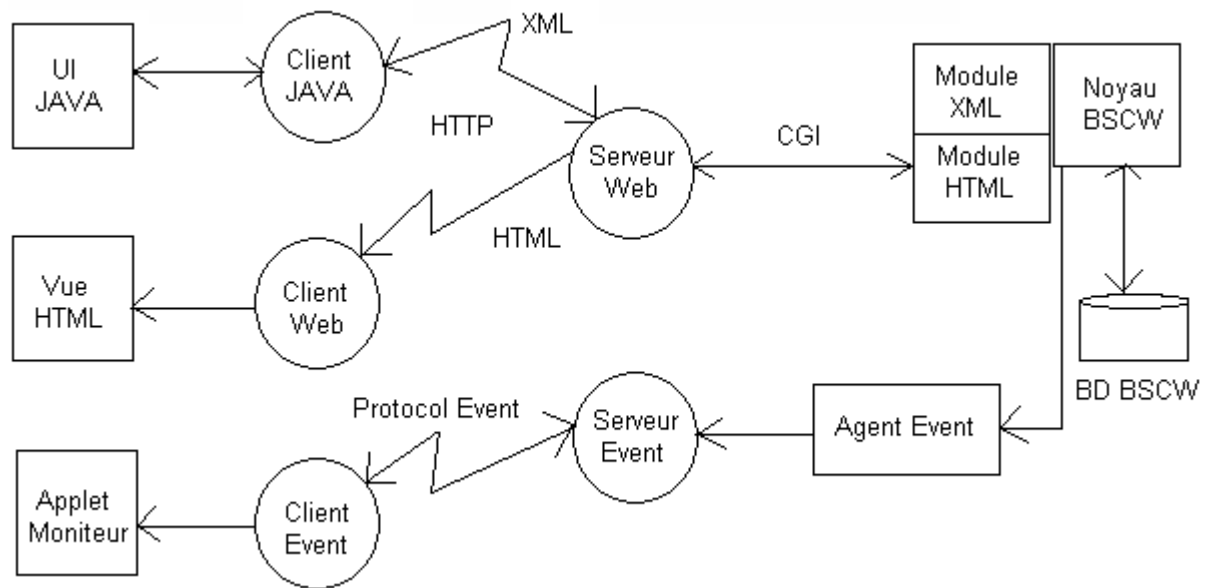


Figure 39 : Architecture du système BSCW (Bentley et al., 1996).

La figure 39, représente une vue d'ensemble de l'architecture du système BSCW. L'interface principale entre le serveur BSCW et les clients (Navigateurs de Web ordinaires) est le protocole HTTP et le langage HTML. Puisque le langage HTML n'est pas très puissant pour respecter la conception d'interface, le système est doté d'une interface additionnelle basée sur le langage Java (utilise XML) qui a été ajouté à la version 3.3 du système en juin 1999.

En plus du serveur BSCW, c.-à-d., un serveur Web étendu avec la fonctionnalité de BSCW, le système BSCW contient un serveur d'événement qui alimente l'applet moniteur par une applet Java qui peut être lancé à partir d'un espace de travail de BSCW avec des événements au sujet de la présence et des activités d'autres utilisateurs du système. C'est un serveur séparé dont la fonctionnalité ne peut pas être ajoutée à un serveur Web ordinaire puisque le protocole HTTP et le langage HTML sont insuffisants pour réaliser ces dispositifs particuliers.

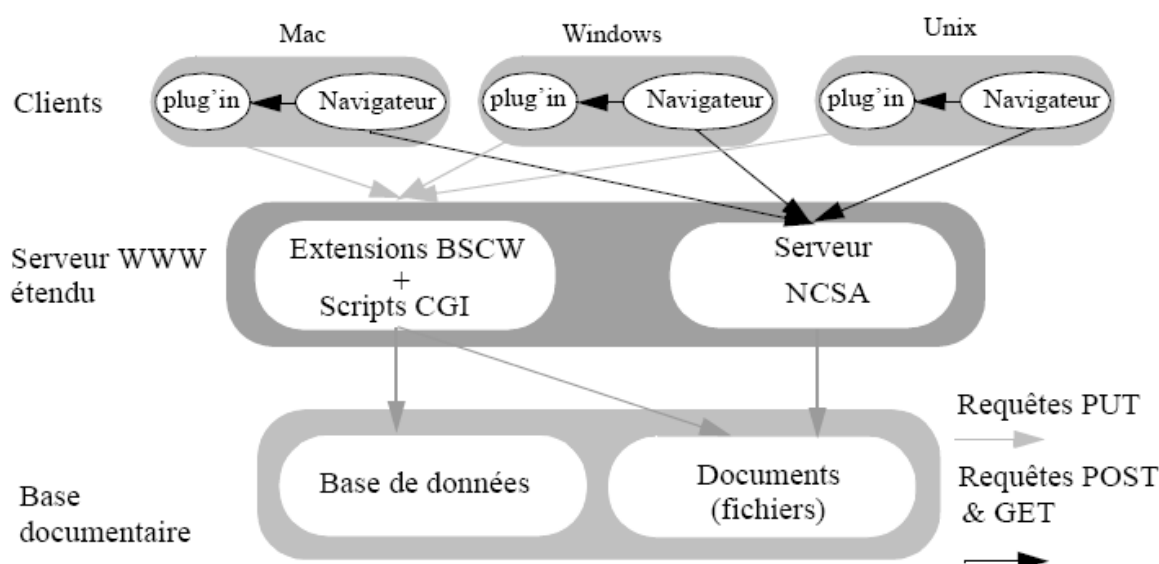


Figure 40 : Environnement d'exécution du système BSCW (Bentley et al., 1996).

Le système BSCW s'exécute sous Windows NT, Mac et sous diverses formes d'Unix (y compris Sun Solaris et Linux) comme le montre la figure 40. Il peut être utilisé avec les serveurs WEB tel que le serveur de Microsoft (IIS), le serveur d'Apache, le serveur d'AOL et les serveurs de CERN.

3.2 Les fonctionnalités du système BSCW

La métaphore principale du système BSCW est l'espace de travail partagé. En général un serveur BSCW contrôlera des espaces de travail pour différents groupes. Les utilisateurs peuvent être des membres de plusieurs espaces de travail. En outre, les utilisateurs peuvent installer des espaces de travail privées qu'ils ne partagent pas avec d'autres.

Un espace de travail partagé peut contenir différents types d'information tels que des documents, des images, des liens (URL) à d'autres pages WEB ou des sites FTP, des discussions entre les utilisateurs des informations sur d'autres utilisateurs, etc. L'espace de travail est organisé en une structure hiérarchique de dossiers. Lors de l'accès à l'espace de travail, son contenu est présenté sous forme d'une page HTML.

Un système coopératif doit fournir des informations d'awareness pour permettre à ses utilisateurs de coordonner leur travail. Les services d'événement du système BSCW avertissent les utilisateurs de tout nouvel événement sur les activités des autres utilisateurs, en ce qui concernant les objets dans un espace de travail partagé.

Des événements sont déclenchés à chaque fois qu'un utilisateur effectue une action dans un espace de travail tel que le transfert d'un nouveau document, téléchargement d'un document existant, renommer un document, etc. Le système enregistre les événements et présente ces événements à chaque utilisateur dans diverses formes, par exemple, comme icônes d'événements attachés aux objets, par l'intermédiaire de l'Email ou par des messages dans l'applet moniteur.

La manière la plus commune d'informer les utilisateurs sur les événements est par les icônes d'événement attachées aux objets. Une telle icône indique qu'un événement particulier s'est produit récemment. Ceci signifie que les événements qui se sont produits pour un objet depuis que l'utilisateur a pour la dernière fois effectué l'action «Au Courant», cette opération est utilisée pour indiquer au système qu'on est au courant des événements qui se sont produit jusqu'ici et qu'on ne souhaite plus les voir (c.-à-d., leurs icônes d'événement) dans l'espace de travail. Les événements peuvent être captés à différents niveaux, de différents objets pour accomplir les structures hiérarchiques des dossiers d'espace de travail. Les histoires d'événement sont, bien sûr, personnel pour chaque utilisateur particulier, par exemple, un événement peut être nouveau à un utilisateur mais vieux pour d'autres utilisateurs.

Le système distingue cinq types d'événements qui sont représentés par cinq différentes icônes d'événement en conséquence :

- Les événements de création indiquent qu'un objet a été créé depuis que l'utilisateur a pour la dernière fois exécute l'action «Au Courant».
- Les événements de lecture indiquent qu'un objet a été téléchargé ou lu par quelqu'un.
- Les événements de changement indiquent qu'un objet a été modifié. Cette catégorie inclue plusieurs types d'événement, tels que l'édition, changement de nom, etc.
- Les événements de mouvement indiquent qu'un objet a changé sa localité. Cette catégorie inclue les événements de la suppression et de l'annulation de suppression et les événements de coupage et d'omission.
- Les événements de touche sont affichés pour montrer, que quelque chose est arrivée à un objet contenu à l'intérieur d'un récipient tel qu'un dossier (dans la racine ou l'un de ses sous-dossiers).

Chaque entrée d'événement décrit ce qui a été fait, quand et par qui. Bien que cette approche pour offrir le sentiment de la présence dans le groupe semble très simple à première vue, L'information telle que «l'utilisateur a téléchargé une nouvelle version du document X », ou «que l'utilisateur B a lu le document Y » est souvent très utile pour que les membres de groupe

coordonnent leur travail et peuvent avoir une vue d'ensemble de ce qui s'est produit depuis qu'ils se sont pour la dernière connectés au serveur BSCW. En outre, le système contient les principales fonctions suivantes :

- **Authentification.** Les utilisateurs ont la possibilité de s'identifier auprès du serveur ; cette identification est utile pour l'attribution de droits d'accès et l'historique des opérations.
- **Gestion de versions et verrouillage.** Les fichiers déposés sur le serveur peuvent être versionnés (de façon linéaire ou arborescente) et verrouillés (seuls le propriétaires détient le droit de modifier le fichier).
- **Forums de discussions.** Ce sont des objets de BSCW qui contiennent des messages textuels ; les messages sont organisés dans une structure arborescente où chaque message peut donner lieu à plusieurs réponses (situées à un niveau inférieur de la structure, le nombre de niveaux n'étant pas limité). Les forums de discussions sont des objets indépendants ou ils sont attachés à un fichier donné (annotation).
- **Droits d'accès.** Ils permettent de restreindre l'accès à certains fichiers ou certains répertoires pour des utilisateurs déterminés. Cette gestion des droits est dans BSCW très complexe d'utilisation (ex. héritage de droits, surcharge de droits, définition et modifications de catégories de droits) et donc potentiellement puissante.
- **Notification d'événements nouveaux.** BSCW compte sept types d'indicateurs d'événements nouveaux ; ce sont de petites icônes qui apparaissent à côté du nom de l'objet (répertoire ou fichier) : (1) lors de la création d'un répertoire ou d'un fichier, (2) lors d'une modification effectuée à l'intérieur d'un répertoire, (3) lors de la modification d'un fichier, (4) lors du renommage d'un fichier, (5) lors de la lecture d'un fichier, (6) lors du blocage d'un fichier et (7) lors du verrouillage d'un fichier.
- **Fonctions de recherches.** Elles sont destinées à retrouver des objets (ex. fichiers, personnes, réunions) de BSCW, principalement des fichiers, selon leur nom et/ou leurs attributs (ex. auteur, date de modification).
- **Compression de fichiers.** BSCW permet de compresser et de décompresser (au format zip) directement sur son serveur une liste de fichiers. Cela permet d'accélérer substantiellement les temps de dépôt et chargement de fichiers. Par exemple, l'utilisateur choisit de charger plusieurs fichiers depuis le serveur ; il l'est fait compresser sur le serveur et charge ensuite le fichier compressé (appelé archive) ; une fois sur sa machine locale, il peut décompresser cette archive et récupérer les fichiers qui la constituent.
- **Planification de réunion.** Il est possible de planifier des réunions, et pour cela, de créer un objet spécifique (au même titre qu'un fichier) qui permet de déclarer des

participants, un ordre du jour, un lieu et une date. Les participants sont ensuite alertés par courrier électronique de la tenue de la réunion et peuvent répondre présent ou absent ; la liste des personnes présentes ou absentes est mise à jour automatiquement et reste visible.

- Accès anonyme. L'accès anonyme permet à des utilisateurs occasionnels de visiter un projet et d'accéder (en lecture seulement) à des fichiers déposés par des utilisateurs enregistrés.
- Personnalisation de l'interface. Le collecticiel s'affiche dans une fenêtre de navigateur. BSCW permet d'exploiter certaines fonctions de navigateur (ex. contrôle ActiveX d'Internet Explorer pour des « glisser-déposer ») pour en améliorer l'utilisation (ex. glisser-déposer). Il est également possible de définir des niveaux d'utilisation (ex. expert, débutant) qui agissent sur le nombre de fonctions proposés. Enfin, il est possible de changer la langue utilisée, c'est-à-dire non seulement les textes, mais également les icônes.

3.3 Les Systèmes similaires

Quand la première version du système BSCW a été sortie en 1995, c'était le premier collecticiel basé entièrement sur la technologie de Web. A cette époque, d'autres collecticiels commerciaux basés sur des protocoles et des formats privés, exigent des installations logicielles spécifiques. En 1996, BSCW a gagné la récompense européenne d'innovation de logiciel (ESIP'96) pour sa nouvelle approche dans le développement de collecticiel.

Depuis ces dernières années, d'autres collecticiels ont émergé qui sont basés sur la technologie d'Internet et de Web. Cela inclut les systèmes qui ont été développés à partir de zéro tel que Hyperwave ou Livelink, mais également les systèmes qui ont remplacé {plus ou moins complètement leur protocole et format privés par des normes ouvertes telles que les versions les plus récentes de domino et de lotus.

Nous pensons que le système BSCW reste toujours un des principaux systèmes qui supportent la collaboration. Sa force est sûrement basée sur la grande rétroaction de sa communauté d'utilisateur qui a contribué beaucoup à l'état actuel du système.

Le système des espaces de travail partagés BSCW est un outil de TCAO sur le WEB offrant une large gamme de dispositifs pour supporter la collaboration et la coopération au sein des groupes. En particulier, le système est considéré comme un outil très utile pour la coopération entre groupes géographiquement dispersés et ceux utilisant des plate-formes différentes.

4 Expérimentation

4.1 Projet et cadre de l'expérimentation

L'expérimentation a été menée dans le cadre d'un projet de fin d'étude en cinquième année informatique industrielle à l'université de Ouargla. L'intitulé du thème était "Construction d'un modèle de simulation pour l'évaluation des performances d'un système de production : Cas de l'entreprise nationale des gaz industriels ENGI". Deux étudiantes, suivant leur encadreur avec la collaboration de deux autres : un ingénieur spécialiste dans la production des gaz (co-encadreur) et un mathématicien, ont participé à l'expérimentation. La durée du projet, et donc de l'expérimentation, était de six mois. La description du projet est comme suite :

«La compétition internationale auxquels se livrent les industriels les conduit à mettre en œuvre des systèmes automatisés complexes pour lesquelles les méthodes traditionnelles de conception sont limitées à cause de leur incapacité à prendre en charge l'outil de production. Aujourd'hui, les entreprises n'ont plus le droit à l'erreur. Faire de mauvais investissements lors de l'achat de machines, sur dimensionner les lieux de stockage, répartir les machines dans l'atelier de manière irréfléchie, toutes ces actions sont actuellement considérées comme appartenant à la préhistoire et complètement indignes ! C'est pour cela que les projets de simulation interviennent, pour dire non aux actes aléatoires et irréfléchis et oui aux actes planifiés et étudiés.

L'approche par simulation constitue une bonne alternative à l'expérimentation pour évaluer les performances des systèmes de production, dans un monde de planification et de prédiction, où tout est calculé et prédéfini à l'avance car rien n'est laissé au hasard. En détaillant précisément le comportement d'un système de production, la simulation met en évidence les dysfonctionnements les plus manifestes. L'analyse spécifique des flux (de matière, de composants, de produits) fournit aux responsables d'ateliers de précieuses indications quant à l'adéquation de la répartition "géographique" des ressources dans le système de production. Elle permet également d'évaluer les charges attribuées aux différents postes de travail et machines et, en cas d'engorgement, d'étudier les potentialités de résolution des problèmes de surcharge.

Dans ce contexte, les travaux du projet seront menés dans le cadre de la simulation pour des intérêts et des perspectives industrielles. Dans cette orientation, la simulation permettra d'une part l'intégration de modèles et scénarios de teste applicatifs et d'autre part servira d'outil d'aide à la prise de décision, de mise au point et d'évaluation des performances. Le travail demander dans ce projet ayant pour but de proposer une construction d'un modèle de simulation pour l'évaluation des performances d'un système de production : cas de l'Entreprise Nationale du Gaz Industriel ENGI. Ouargla"»

4.2 Objectifs de l'expérimentation

Les expérimentations d'outils logiciels coopératifs dans le domaine de simulation sont rares; les seules expérimentations dont nous avons eu connaissance ont des objectifs différents des notre; les hypothèses qui les sous-tendent sont quelque peu différentes de celles que nous essayons d'étudier : L'aspect coopératif d'un projet de simulation justifiant la prise en compte de la dimension de groupe dans les outils de simulation. Les objectifs de notre expérimentation sont :

- Instancier un collecticiel de simulation dans le but de valider les modèles d'un collecticiel de simulation élaborés dans le chapitre précédent.
- De mieux connaître les besoins d'une équipe de projet de simulation en terme d'outils de coopération.
- la compréhension global du projet au travers de l'outil : comment les membres d'un projet de simulation se représentent-ils l'avancement du projet, les tâches qu'ils ont à conduire et celles que leurs partenaires ont à conduire?

4.3 Méthodes

Tout projet coopératif requiert de la part des participants, dès son démarrage, un savoir méthodologique commun. Nous nous sommes employés à en fournir un, sous la forme d'un document que nous avons appelé guide de projet. Dans ce guide, nous essayer de faire la distinction entre le niveau de l'expérimentation (production et exploitation des résultats) et le niveau du projet de simulation (production de modèles et de mesures), nous avons été amené à définir à la fois une méthode à destination des étudiants, concepteurs du projet et une méthode à destination des expérimentateurs.

Notre guide contient plusieurs points répondant à des exigences diverses :

- *Identité et rôles des membres.* La liste des participant au projet est établie : les principaux membres sont les deux étudiants en informatique industrielle ; ce sont eux qui doivent réaliser le projet de simulation, suivi dans cette tâche par leur encadreur qui joue le rôle de chef de projet. Hormis ces trois membres directement impliqués, trois autres membres, un co-encadreur délégué par l'entreprise, le responsable du service de production dans l'entreprise et un mathématicien.
- *Phasage.* Le projet est décomposé en trois grandes étapes menées de façon collective : Analyse du problème, construction de la simulation et l'exploitation de la simulation. La première étape permet de préciser le contexte de l'étude. Elle comprend l'identification du problème; la spécification des objectifs, la réalisation d'une première ébauche du modèle qui a pour but d'en délimiter les frontières et de spécifier les données dont on a besoin et la validation de ce modèle auprès de l'utilisateur. Le but à atteindre dans cette étape est de construire un modèle valide tout en restant cohérent avec les objectifs de l'étude. L'étape de construction d'un modèle comprend la modélisation logico-mathématique qui peut être facilitée par un outil graphique, et la programmation proprement dite. Dans sa première itération, cette étape se termine par une validation qui consiste à comparer le comportement du modèle avec celui du système réel qu'il est censé représenter. Dans la troisième étape, le modèle validé peut servir à l'évaluation du comportement dynamique du système. Cette étape nécessite une définition précise de la campagne d'exploitation (quelles hypothèses veut-on vérifier, dans quel contexte), la production de mesures par la simulation proprement dite, la mise en forme et la comparaison des résultats obtenus aux objectifs poursuivis. S'ils n'ont pas été atteints, de nouveaux scénarios sont proposés et testés jusqu'à satisfaction. Dans la mesure où la plupart des modèles comportent des aléas, l'étape d'exploitation nécessite que soient déterminés avec rigueur la durée de la simulation et le nombre de répliques (exécution du modèle de simulation); elle fait appel aux outils statistiques afin de caractériser le comportement du modèle : calcul d'intervalles de confiance, de coefficients de corrélation, ...
- *Format de fichier.* Pour faciliter la relecture de fichiers, quelques formats de fichiers sont imposés : le format doe (Arena) pour la construction des modèles, le format JPEG pour l'image par points, le format RTF pour les documents.
- *La structure des répertoires :* chaque membre dispose d'un répertoire personnel, sorte de boîte à lettres, à partir de laquelle il communique personnellement avec les autres

membres du projet. Ces répertoires sont regroupés par famille de membre (ex. Chef de projet, Client de l'étude, les informaticiens, les ingénieurs de métiers, les utilisateurs); lorsque les communications concernent l'ensemble des acteurs d'une famille donnée, c'est à ce niveau qu'elles s'établissent. Le répertoire racine contient tous les répertoires par famille de membre ; les objets qui y sont entreposés concernent donc tous les membres du projet. Notre expérimentation ne fait pas usage de la gestion des droits d'accès (proposée par BSCW) pour rendre visibles ou non certains objets en fonction de l'identité de l'utilisateur; c'est la localisation de l'objet dans l'arborescence des répertoires qui détermine son auditoire.

- *Formes des requêtes.* Dans cette expérimentation les participants ont la possibilité de communiquer entre eux, leur question, leur approbation, etc, par le biais des fils de discussion, des courriels ou bien directement par les moyens offerts par les outils de vidéo conférence et de partage des applications. Lorsque celles-ci ont un caractère générale, elles doivent être créées en tant que fil de discussion indépendant, à l'intérieur de la hiérarchie des répertoires; la position qu'elles y occupent étant directement liée, selon la règle convenue, à l'auditoire qu'elles sont censés atteindre.
- *Répertoires publics.* Pour éviter des procédures d'enregistrement auprès du serveur BSCW, un répertoire public a été mis en place dont l'accès anonyme, bien que limité à certaines opérations, est le moyen le plus simple de présenter des documents aux membres extérieurs (Experts consultants). Seuls y sont présentés des documents dont l'intérêt et le niveau de définition sont jugés utiles.

4.4 Limites de l'expérimentation

Toute expérimentation, surtout dans un domaine où les composantes humaine et sociale sont si fortes, présente des limites dans la généralisation de ses conclusions. Citons au moins deux raisons : les données contextuelles de notre expérimentation sont très spécifiques (nombre des participants à l'expérimentation, relations entre les participants, nature du projet, etc.); elles illustrent une diversité combinatoire que seules plusieurs expérimentations pourraient approcher. D'autre part, la nature même de l'expérimentation motive les participants qui consentent des efforts sans doute inaccoutumés dans une pratique ordinaire.

En plus de ces deux facteurs, notre expérimentation est sujette à des limitations qui lui sont propres et qui reflètent le décalage entre conditions expérimentales et conditions réelles. Il convient de les préciser :

- Le projet mené par les étudiants dans un contexte pédagogique bien différent du contexte réel. Les étudiants présentent fréquemment à leur encadreur l'avancement de leur travail. L'expérimentation du collecticiel ne pouvait interférer avec ce projet pédagogique; il n'était en particulier pas question de confronter les étudiants à l'utilisation d'un collecticiel et aux éventuels difficultés techniques susceptibles de survenir.
- Les conditions privilégiées d'accès à Internet, en ce qui concerne la vitesse de transferts et la connexion permanente, favorise une utilisation dont la fréquence n'a sans doute pas d'équivalent dans le monde des professionnels d'aujourd'hui; les étudiants n'ont donc aucune peine à lire et à déposer fréquemment des fichiers volumineux. Là encore, pour minimiser ce décalage, nous pouvons considérer que les conditions de connexion sont appelées à évoluer pour le grand public, et rejoindre celles de l'expérimentation.
- La présence fréquente dans un même lieu et au même moment des étudiants n'était pas de nature à encourager la communication via le collecticiel. Il est plus naturel de s'adresser oralement à une personne que l'on a face à soi plutôt que de recourir à une interface logicielle.

4.5 Déploiement du projet de simulation

La naissance électronique du projet sur le système BSCW a fait l'objet d'une réservation d'un espace logique administré par le chef de projet (encadreur) et d'un espace physique correspondant. Les participants au projet s'abonnent à cet espace logique. Ce dernier est organisé selon une structure arborescente de répertoires et de fichiers, bien connue et c'est la raison principale sans doute de son choix par les utilisateurs en informatique. Dans notre expérimentation cette structure en répertoires a été utilisée pour structurer l'espace logique (ex: espace commun, espaces des groupes, espaces privés des membres) et la création des artefacts du projet.

La philosophie de fonctionnement du système BSCW et ses fonctionnalités nous ont permis d'implémenter parfaitement les deux modes de coopération asynchrone et session. La figure 41 représente l'espace du chef de projet (l'encadreur) qui travail en mode asynchrone. Dans cette

session de travail, le Chef de projet prépare les références bibliographiques nécessaires pour le projet des deux étudiantes. La figure 42 représente l'espace des deux étudiantes en mode session. Dans cette session, les étudiantes consultent leur encadreur via un téléphone SIP.

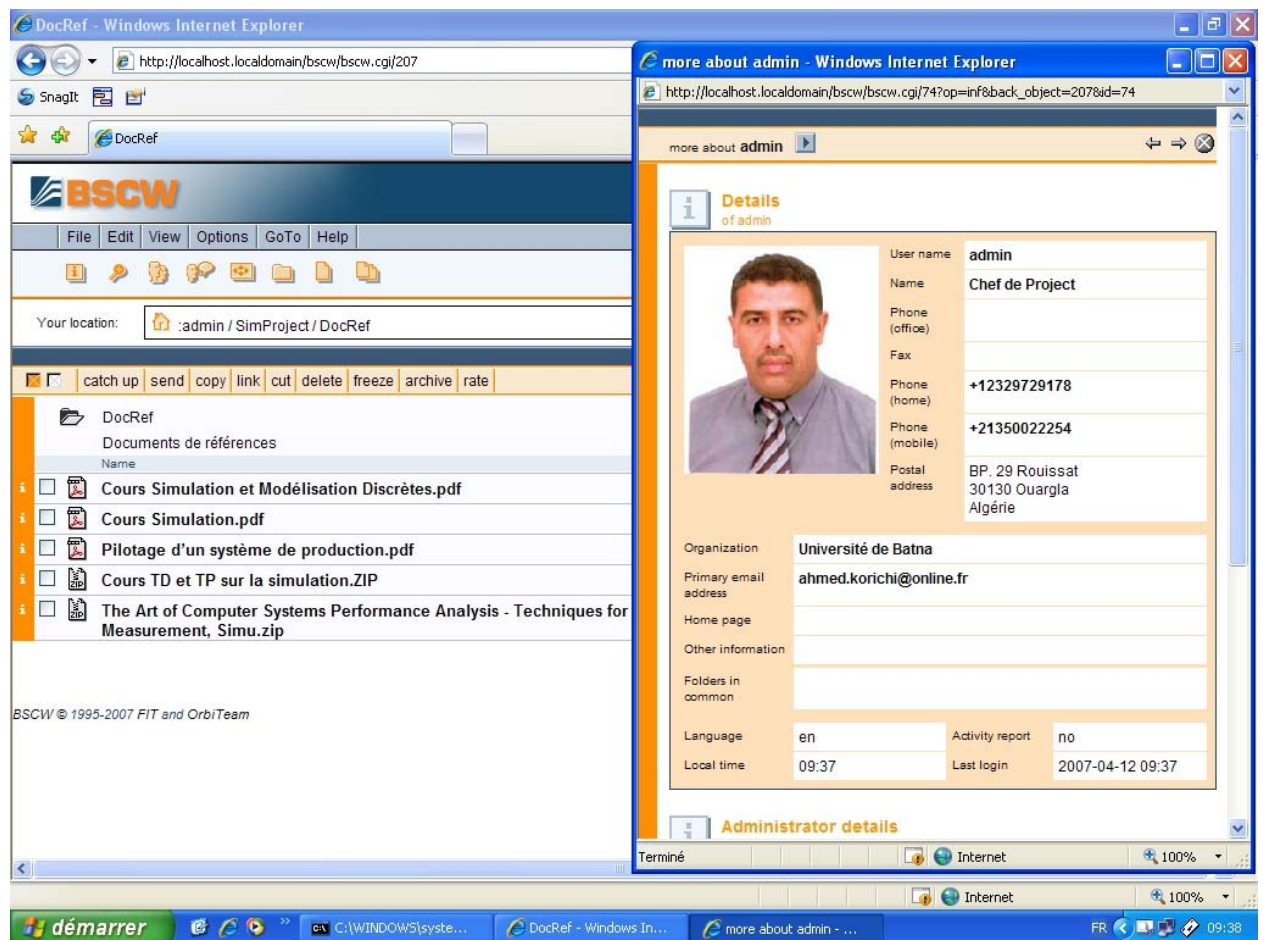


Figure 41 : Espace du Chef de projet qui travail en mode Asynchrone

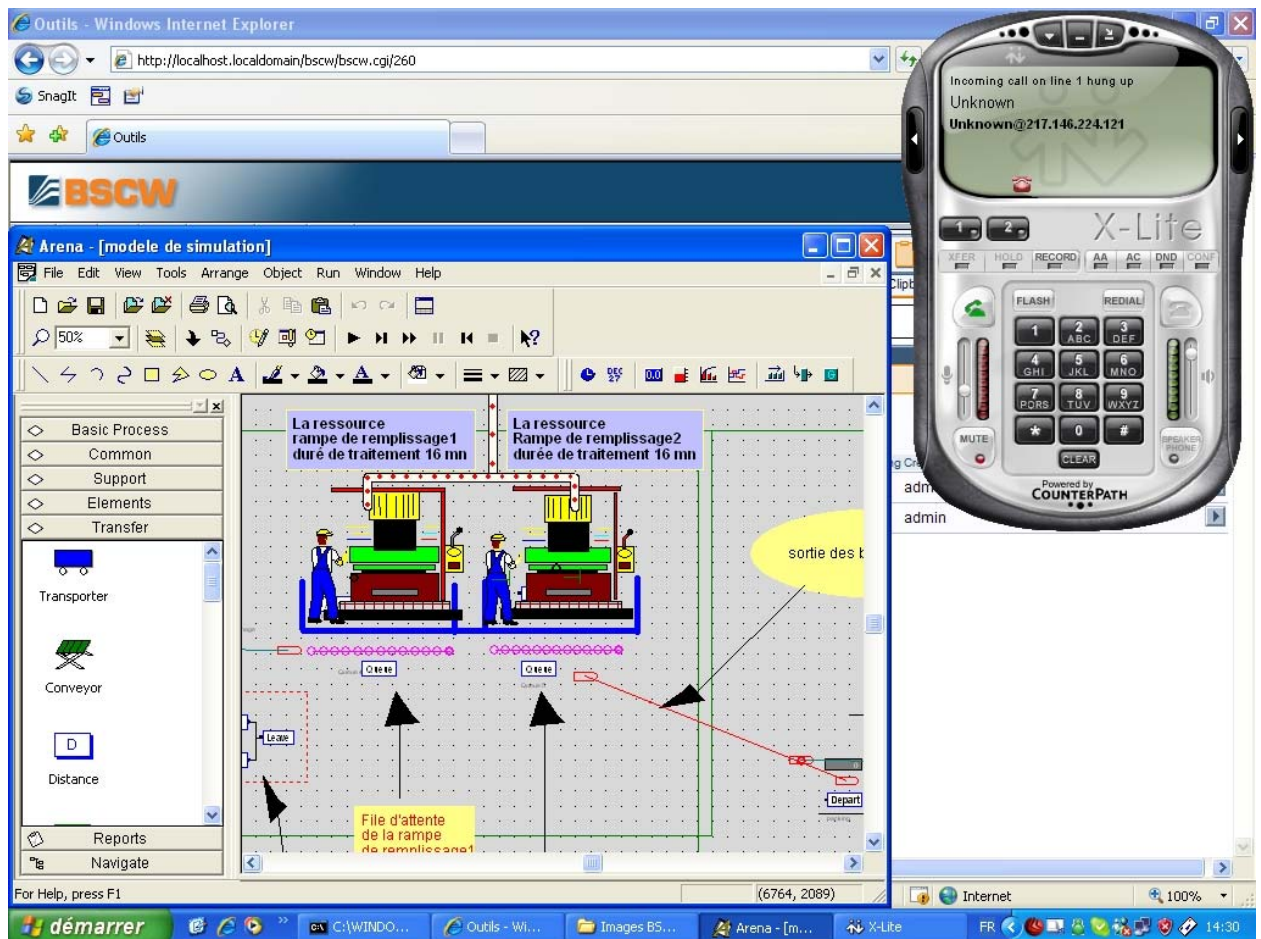


Figure 42 : Espace des étudiants qui communiquent via X-LITE

Pour l'implémentation des deux autres modes de coopération se fait par la possibilité de l'utilisation des logiciels externes appropriés offerte par le système BSCW (ex. NetMeeting). Lors le déroulement du projet, la coordination et le suivi des équipes et des travaux est effectué par le biais de mécanismes de supervision et de travail collaboratif.

La figure 43 représente l'espace d'une étudiante qui construit son modèle avec l'assistance de son encadreur via le logiciel NetMeeting. Dans ce cas, l'encadreur et l'étudiante travaillent en mode étroite. L'application Rockwell ARENA est partagée par le NetMeeting. La figure 44 représente l'espace du chef de projet qui change son profil de communication par l'intégration du logiciel NetMeeting comme outil de communication.

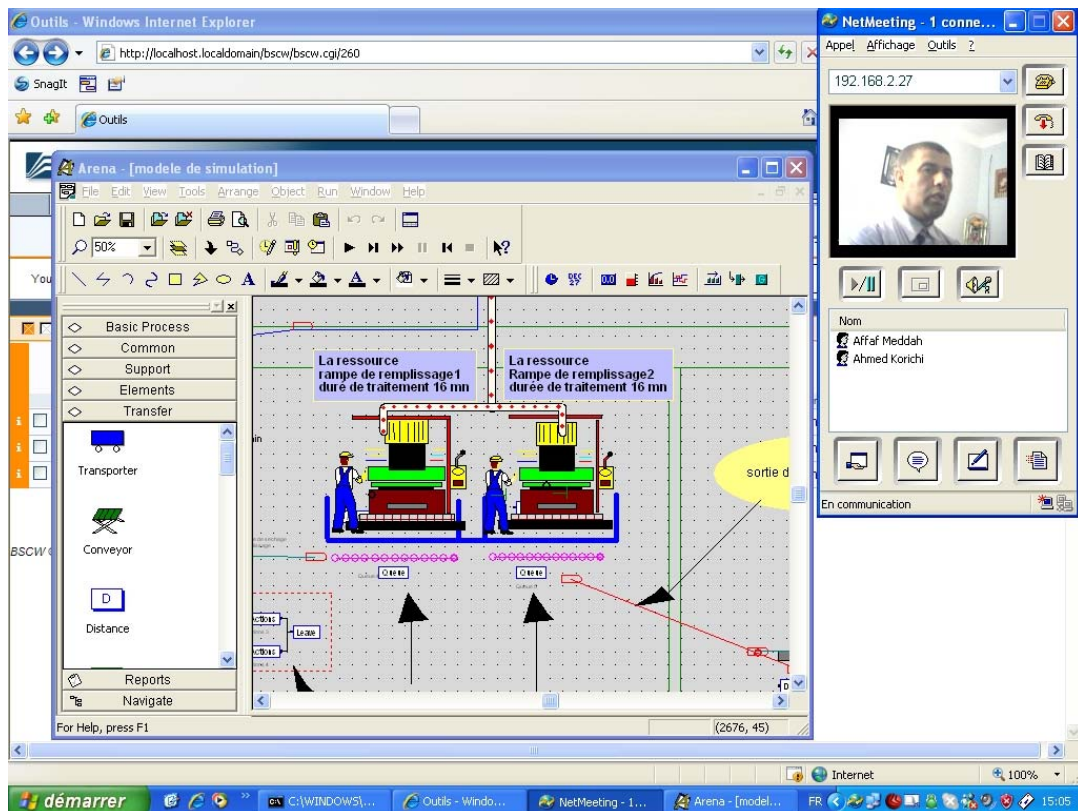


Figure 43 : BSCW & NetMeeting en cours d'utilisation

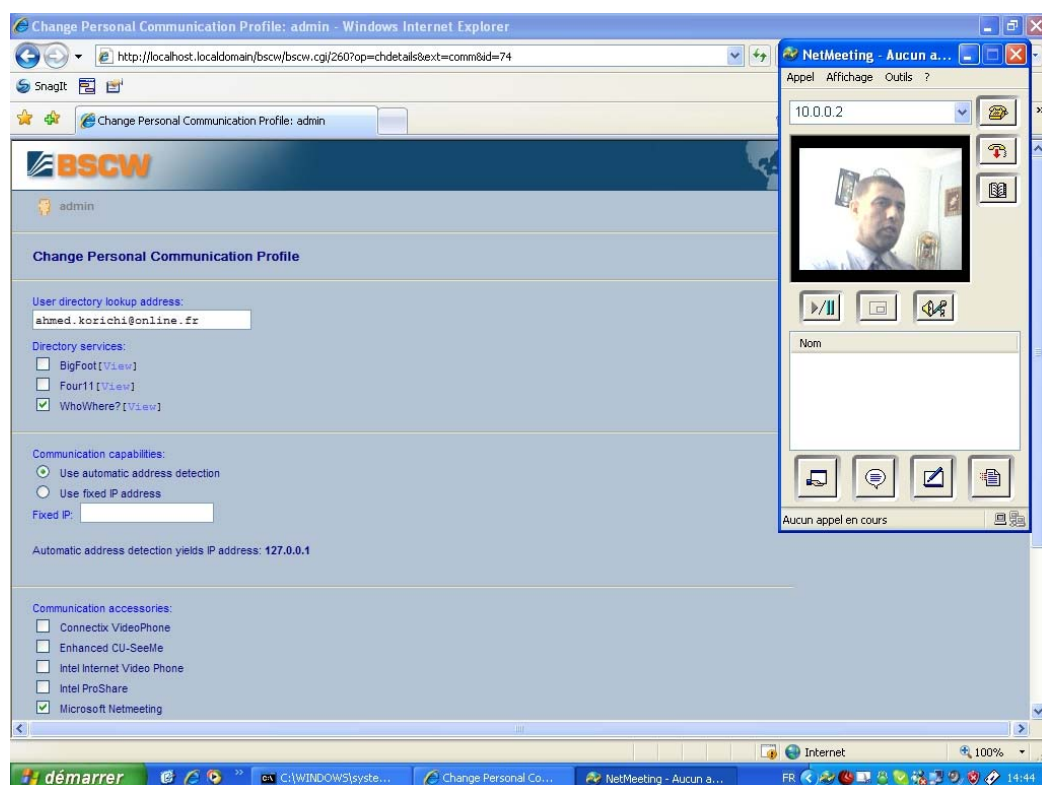


Figure 44 : Changement du profil de communication du chef de projet

5 Résultats

Le résultat final des travaux des deux étudiants dans le cadre de leur projet pédagogique, est la réalisation d'un modèle de simulation pour l'évaluation des performances du système de production de l'ENGI. Ce modèle a permis la compréhension du comportement de ce système pour une configuration donnée. Le comportement d'un système de production est défini par des performances de type : production maximale, temps de réponse, temps d'attente des demandes, ... Ces performances sont calculées pour une configuration donnée par des caractéristiques sur les machines (nombre de machines, taux de panne), sur le routage des pièces, sur l'environnement (arrivée des pièces, des demandes), ...

Comme mentionné ci-dessus la modélisation et la simulation est réalisé par le simulateur SIMAN/Arena, développé par Systemes Modeling Corporation. Il est notamment dédié à la modélisation, simulation et animation de systèmes de production. Le logiciel est basé sur les concepts de programmation orientée objet et de modélisation hiérarchique, utilisant la puissance et la flexibilité de modélisation du système SIMAN/Cinema.

A l'aide d'Arena on peut développer un projet complet de simulation. Arena est un support intégré pour l'analyse des données d'entrée, la construction du modèle, l'exécution interactive, l'animation, la traçabilité et la vérification de l'exécution et l'analyse des sorties.

Les utilisateurs peuvent développer des nouveaux modèles dans la fenêtre du modèle, des nouvelles constructions de modélisation dans la fenêtre de template, identifier les caractéristiques des données d'entrée dans la fenêtre d'entrée et analyser les résultats de simulation dans la fenêtre de sortie.

Malgré la complexité du système étudié, le logiciel ARENA nous a permis de construire un modèle très proche de réalité. Ce modèle intègre plusieurs fonctionnalités et fiches d'aide pour faciliter et simplifier sa compréhension. La figure 45 représente la simulation du processus de remplissage des bouteilles de gaz.

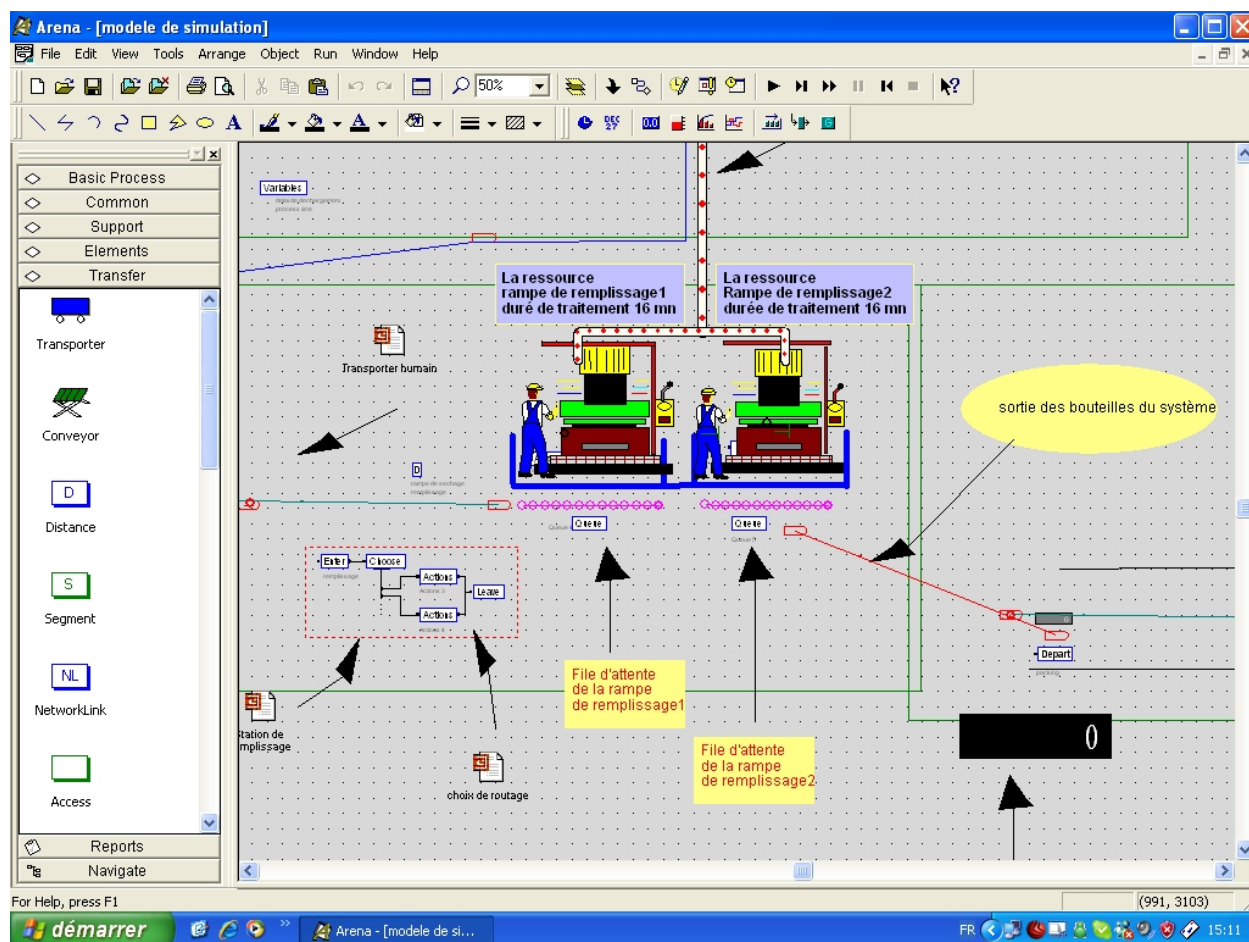


Figure 45 : Simulation du processus de remplissage des bouteilles de gaz.

Bien que l'utilisation du système BSCW permette de concrétiser la conduite d'un projet de simulation coopératif, il était nécessaire de faire un sondage pour connaître les limites de cette solution. Ceci nous a amené à faire une série de démonstrations suivies par un questionnaire.

La Figure 46 représente les résultats du sondage. Il a été demandé à chaque participant d'évaluer l'adéquation de l'implémentation et d'attribuer une note sur une échelle de 1 à 5 dans chaque mode de coopération. Il a été aussi demandé aux participants d'estimer leur compétence en simulation et en technologie d'information. Bien que l'échantillon des participants soit petit, le résultat de cette expérience est extrêmement clair.

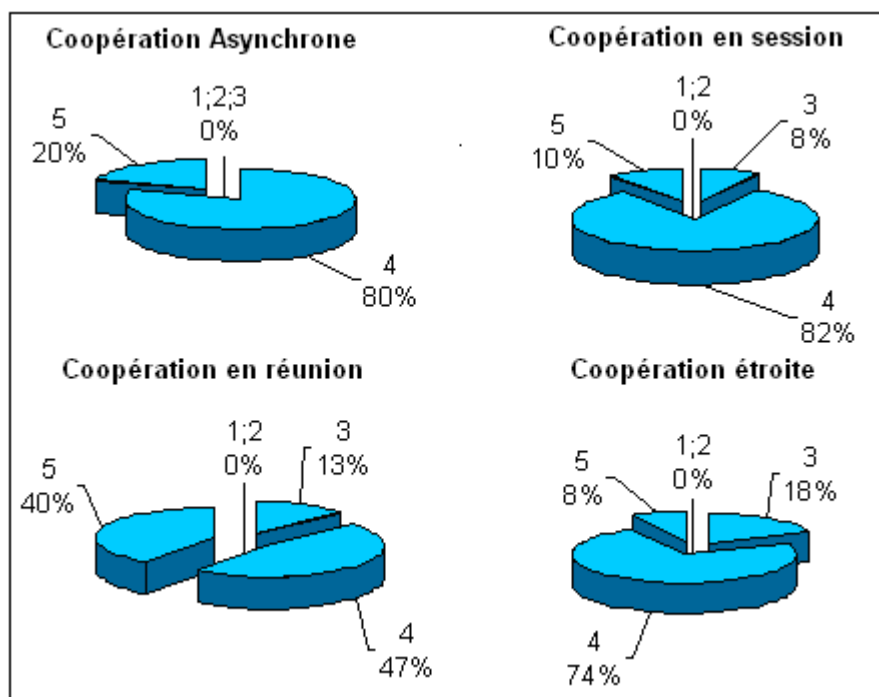


Figure 46 : Résultats de sondage

6 Conclusion

La modestie de notre expérimentation nous empêche d'être absolument formel quant à des exigences que nous pourrions formuler concernant un collecticiel de simulation. Il faudrait répéter ce genre d'expérimentation en faisant varier les données contextuelles pour tracer le contour d'une validation plus fiables de nos modèles.

Chapitre n° 6

Conclusion

VI Conclusion

Pour terminer ce mémoire, nous faisons ici le bilan de l'étude réalisée. Nous résumons les apports, les limitations de ce travail et nous dégagons un certain nombre de perspectives de recherche pour la poursuite de l'implémentation d'un collecticiel de simulation orienté vers l'analyse et la conception en groupe des systèmes de production.

1 Bilan de l'étude réalisé

L'objectif du travail présenté dans ce mémoire vise à explorer l'apport du TCAO dans la conduite d'un projet de simulation, ceci dans le but d'adopter un regard nouveau dominé par l'inéluctabilité technologique actuelle sur les outils de simulation existants. Plus précisément, ce travail vise à spécifier et développer un collecticiel de simulation dans le but d'ajouter la dimension de groupe dans les outils de simulation. Ce collecticiel est principalement destiné à assister les membres d'un projet de simulation dans leurs activités coopératives tout au long de la durée de vie du projet. Les retombées d'un tel environnement sont inestimables tant sur le plan organisation du travail que sur la qualité et les délais de réalisations.

Dans ce cadre, nous avons tout d'abord étudié les méthodes issues du domaine de simulation afin de mieux comprendre les fondements théoriques ainsi que les outils de mise en œuvre sur ordinateur. Ensuite, nous avons étudié les travaux ayant touché de près ou de loin notre axe de recherche. En conclusion, cette étude nous a permis de faire un bilan de synthèse des orientations des recherches dans le domaine de la simulation en générale et la simulation coopérative en particulier.

Cette première partie de notre étude nous a permis :

- D'une part, de mettre en évidence que tout projet de simulation, quelque soit son envergure, requière des compétences variées et ne peut en aucun cas être considéré comme l'œuvre d'une personne isolé mais plutôt celle d'un groupe. Ceci nous a amené à conclure sur la nécessité d'une prise en compte de la dimension coopérative dans les outils de simulation.
- D'autre part, de cerner les éléments d'analyse du contexte spécifique à ce problème et de mieux situer notre travail.

En se basant sur les résultats de la première partie de cette étude, et en vue de proposer un modèle de collecticiel de simulation pour mieux cerner les attentes des utilisateurs, nous avons étudié les collecticiels et leurs outils de conception et de développement logiciels. Nous avons examiné essentiellement les travaux de recherche issus du domaine du travail coopératif assisté par ordinateur, selon deux points de vue différents : Tout d'abord, l'état de l'art des outils de conception et de développement logiciels des applications de groupe en vue de choisir celle qui conviendrait le mieux en prenant en compte les objectifs visés par cette thèse. Ensuite, l'état de l'art des offres du TCAO adéquates pour la mise en œuvre d'un collecticiel de simulation.

Après avoir étudié les collecticiels et leurs outils de conception et de développement logiciels, nous avons mis en application ces méthodes dans notre cas. Cette mise en application a été réalisée au cours de plusieurs étapes. Tout d'abord, en appliquant le modèle de Denver, nous avons dégagé les caractéristiques d'un collecticiel de simulation, du point de vue des cinq critères de ce modèle : Personnes, Artéfacts, Tâches et activités, Situation d'interaction et Protocole social de l'interaction. Cette caractérisation a été validée par une publication dans une revue spécialisée. Nous avons ensuite cerné les besoins fonctionnels et ergonomiques du collecticiel visé. L'utilisation du modèle de Denver nous a facilité la tâche pour élaborer un modèle de tâches d'un projet de simulation coopérative en utilisant la méthode GTA.

Après avoir caractérisé notre collecticiel, nous nous sommes intéressé à l'architecture logicielle d'un tel collecticiel. Pour cela, nous avons examiné la possibilité d'utiliser un modèle parmi ceux existants en l'adaptant à notre domaine si nécessaire. Nous nous sommes servis des résultats d'une étude menée par Y. Laurillau (Laurillau, 2002) sur les modèles d'architecture pour les collecticiels. Notre choix s'est porté sur le modèle de Dewan. Ce choix est motivé principalement par les résultats de cette étude.

Conventionnellement la partie expérimentale est placée à la suite des parties consacrées à la modélisation et à la création de prototype (dans le but de valider ces dernières, d'en évaluer la pertinence et la performance, et de comparer une proposition nouvelle à des propositions antérieures qu'elle est censée améliorer). Dans notre nous avons utilisé un collecticiel prêt à l'emploi pour conduire une série d'expérimentation. Nous avons choisi le système BSCW comme collecticiel d'expérimentations. Ce choix est justifié principalement à cause de l'importance du BSCW dans le domaine du TCAO, de sa simplicité, de son universalité et de sa disponibilité gratuite pour la communauté pédagogique.

La philosophie de fonctionnement du système BSCW et ses fonctionnalités nous ont permis d'implémenter parfaitement les deux modes de coopération asynchrone et session. L'implémentation des deux autres modes de coopération se fait uniquement par la possibilité de l'utilisation des logiciels externes appropriés (ex. NetMeeting). Une expérimentation a été menée dans le cadre d'un projet pédagogique intitulé "Construction d'un modèle de simulation pour l'évaluation des performances d'un système de production : Cas de l'entreprise nationale des gaz industriels ENGI". Bien que l'utilisation conjointe de BSCW et de NetMeeting permet de concrétiser la conduite d'un projet de simulation coopératif, il était nécessaire de faire un sondage pour connaître les limites de cette solution. Ceci nous a amené à faire une série de démonstrations suivies par un questionnaire. Les résultats de cette expérience sont extrêmement intéressants.

2 Apport

L'apport de notre contribution résulte de l'approche pluridisciplinaire que nous avons adoptée tout au long de cette étude. Nous avons ainsi cherché à mettre en correspondance les connaissances que nous avons pu acquérir concernant le domaine de la simulation avec des travaux menés dans les domaines des sciences sociales et du TCAO. D'un point de vue théorique, les modèles que nous proposons dans ce mémoire permettent de préciser et de détailler l'adaptation des résultats des recherches issus du domaine du TCAO au domaine de simulation. Nous avons orienté nos recherches vers la définition d'un modèle de projet de simulation coopératif, puis vers la spécification de solutions informatiques permettant d'assister les membres d'un projet de simulation dans leurs activités coopératives. Notre recherche a permis, d'une part, de rationaliser la définition d'un collecticiel appliqué au domaine de la simulation en nous rapprochant d'une typologie de modélisation standard, et d'autre part, de proposer une solution technique basée sur les offres actuelles du domaine du TCAO (BSCW)

3 Limitations et extensions

Au cours de ce travail, nous avons dû faire face à plusieurs difficultés inhérentes au contexte dans lequel nous évoluons, les limitations dont nous faisons état ici en sont la conséquence directe. La nature de cette thèse nous a conduit à explorer plusieurs domaines, tels que les sciences sociales ou les collecticiels afin d'établir des parallèles avec le domaine de simulation. Les difficultés que nous avons alors éprouvées ont été liées à la maîtrise puis à la mise en relation de concepts appartenant à des univers très différents pour construire un ensemble cohérent. Au-delà de ces contraintes, la position que nous avons adoptée, au confluent de plusieurs sciences fut largement profitable en terme d'enseignements et d'échanges.

Un second obstacle concerne la réalisation d'expérimentations dans le cadre de projets réels. Toute expérimentation, surtout dans un domaine où les composantes humaine et sociale sont si fortes, présente des limites dans la généralisation de ses conclusions. Citons au moins deux raisons : D'une part, les données contextuelles réelles sont très variables (nombre des participants à l'expérimentation, relations entre les participants, nature du travail demandé, etc.). Elles illustrent une diversité combinatoire que seules plusieurs expérimentations pourraient approcher. Par conséquent, ils n'ont que peu de chance d'être simulées par une seule expérimentation. D'autre part, la nature même de l'expérimentation motive les participants qui consentent des efforts sans doute inaccoutumés dans une pratique ordinaire. En plus de ces deux facteurs, notre expérimentation est sujette à des limitations qui lui sont propres et qui reflètent le décalage entre conditions expérimentales et conditions réelles.

4 Perspectives

La conception et le développement d'un collecticiel, quel que soit son domaine d'application, est certainement une tâche ambitieuse, en particulier vu le temps imparti d'une thèse. La problématique présentée dans ce mémoire ouvre plusieurs perspectives de recherche concernant la création de modèles et les applications dédiées au domaine de la simulation coopérative. Nos perspectives à court terme visent particulièrement à compléter ce travail du point de vue architecture logiciel et implémentation.

Par ailleurs, l'approche que nous avons employée pour la création des modèles que nous avons présentés semble être suffisamment flexible pour permettre des évolutions et des développements futurs. En effet les modèles que nous avons proposés peuvent être largement exploitées dans un autre créneau, comme celui de l'enseignement coopératif de la simulation.

*Références
Bibliographiques*

VII Références bibliographiques

- [Andrew, 1999] Andrew F. Seila, John A. Miller. (1999). Scenario Management in Web Based Simulation. *Proceedings of the 31st conference on winter simulation: Simulation --- a bridge to the future*, Vol. 2, pp 1430-1437.
- [Appelt et Busbach, 1996] Applet, W. and Busbach U. (1996). The BSCW System: A WWW-based application to support collaboration of distributed groups. *Proc. 5th Workshop on Web Infrastructure for Collaborative Applications*, June 1996, Stanford, California. pp. 304-309.
- [Appelt, 1999] Applet, W. (1999). WWW Based Collaboration with the BSCW System. *Proceedings of the Conference on Current Trends in Theory and Informatics*. p. 66-78.
- [Beca et al., 1997] Beca, Lukasz, Cheng, Gang, Fox, Geoffrey C., Jurga, Tomasz, Olszewski, Konrad, Podgorny, Marek, Sokolowski, Piotr, & Walczak, Krzysztof. (1997). Web Technologies for Collaborative Visualization and Simulation. *Proc. of the 8th SLAM Conference on Parallel Processing for Scientific Computing*. Minneapolis, Mar.
- [Belattar et Korichi, 2001a] Belattar, B., Korichi, A. (2001). GOMSE: A Group Oriented Modeling and Simulation Environment. *ACS/IEEE International Conference on Computer Systems and applications (AICCSA'2001)*, Beirut Lebanon, June 25-29.
- [Belattar et Korichi, 2001b] Belattar, B., Korichi A. (2001). Intégration des Concepts du Travail Coopératif Assisté par Ordinateur en Simulation. *Conférence Internationale JIM'2001*, Metz France, 4-7 Juillet.
- [Belattar, 2000] Belattar, B. (2000). *Simulation & modélisation*. Support de Cours. Département d'Informatique, faculté des sciences de l'ingénieur, Université de Batna (Algérie).
- [Belattar et al., 1997] Belattar, B., Zidani, A., Djoudi, A. (1997). Vers un Environnement de Simulation Coopératif, *Actes de la conférence MOSIM'97*, Rouen France, 5-6 juin 1997, pp. 483-499.
- [Belattar et al., 1991] Belattar, B., Pierrevaln H., Dussauchoy A. (1991). An intelligent modeling environment based on a graphical interface. *3rd International Symposium on Systems Research Informatics and Cybernetics*, Baden-Baden Germany, August 1991.
- [Benatallah, 1997] Benatallah, S. (1997). *Architectures systèmes pour la construction et l'exécution de collecticiels*. Thèse de doctorat de l'université de Savoie, 27 Juin. 165 pages.

- [Bentley et al., 1997a] Bentley R., T. Hoserstmann, Trevor J. (1997). The World Wide Web as enabling technology for CSCW: The case of BSCW. *The Journal of Collaborative Computing (CSCW)*. Vol. 6, Num 2-3, Kluwer Academic Press, Amsterdam. p. 111-134.
- [Bentley et al., 1997b] Bentley, R., Taschner T., Appelt W. (1997). The Freedom to Choose: Transforming Content On demand in the BSCW Shared Workspace System. *Proceedings of WebNet 97 - World Conference on the WWW, Internet & Intranet*. November 1-5, Toronto, Canada.
- [Bentley et Appelt, 1997c] Bentley, R., Appelt W. (1997) Designing a System for Cooperative Work on the World-Wide Web: Experiences with the BSCW System. *30th Annual Hawaii International Conference on System Sciences (HICSS-30)*. 7-10 January, Maui, Hawaii.
- [Bentley et al., 1996] Bentley, R., Busbach, U. and Sikkel, K. (1996). The Architecture of the BSCW Shared Workspace System. In *5th ERCIM/WG4 Workshop on CSCW and the Web*, Arbeitspapiere der GMD 984, Sankt Augustin, Germany, pp. 31-42.
- [Beuscart et al.,1994] Beuscart, R., Yousfi F., Dufresne E., Derycke A..(1994) Travail coopératif et groupware. In *Informatisation de l'Unité de soins du futur, France, Springer-Verlag*. pp. 195-210.
- [Bourguin, 2000] Bourguin, G. (2000). *Un support informatique à l'activité coopérative fondé sur la Théorie de l'Activité : le projet DARE*. Thèse de doctorat de l'université de sciences et technologies de LILE. 215 pages.
- [Brinzei, 1998] Brinzei N. (1998). *L'état de l'art sur la conduite des systèmes flexibles de production*, Rapport à l'Université Polytechnique de Timisoara.
- [BSCW, 2006] BSCW (Basic Support for Cooperative Work) © GMD (2006). , Site Web accessible à l'url: <http://bscw.gmd.de>.
- [Cubert et al., 1998] Cubert, R. -M. and Fishwick, P. -A. (1998) Software architecture for distributed simulation multimodels. In *SPIE AeroSense98 Conference Proceedings*, SPIE, Bellingham, WA, April 1998.
- [Courbon et Tajan, 1999] Courbon, J.C. et Tajan S. (1999) *Groupware et Intranet Vers le partage des connaissances*. Dunod, 2^{ème} Edition, Paris. 356 pages.
- [Domingo et al., 1998] Domingo, C. Sananes, M. Tonella, G. (1998). Web Modeling-Simulation: Two Strategies. *Simulation Series*, 1998, Vol 30, Num 1, 45-50.

- [Dumas, 1999] Dumas, C. (1999). *Un modèle d'interaction 3D : Interactions homme - machine et homme - machine - homme dans les interfaces 3D pour le TCAO synchrone*. Thèse de doctorat de l'université des Sciences et Technologies de Lille, U.F.R. d'I.E.E.A., 01 Octobre. 203 pages.
- [Ellis et al. 1991] C. Ellis, S. Gibbs et G. Rein (1991). Groupware: some issues and experiences. *Communications of the ACM*, Vol 34, Num 1, Pages 38-58.
- [Ensor and Carraro, 1998] Ensor, J. and Carraro, G. (1998). Peloton: A Distributed Simulation for the World Wide Web. In *Proc. 1998 International Conf. On Web-based Modeling and Simulation*, San Diego, CA, 12-14 January.
- [Erard et Deguenon, 1999] Erard, P.J. et Deguenon P. (1999) *Simulation par événements discrets*. P.P.U.R, 1^{er} Edition, Lausanne 1996. 417 pages.
- [Fishwick, 1996] Fishwick, P.A. (1996). Web-Based Simulation: Some Personal Observations. In: *Proceedings of the 1996 Winter Simulation Conference*, Coronado, CA, 8-11 December. pp. 772-779.
- [Haake et al., 1999] Jörg M. Haake, Uffe K. Wiil and Peter J. Nürnberg. (1999). Openness in shared hypermedia workspaces: The case for collaborative open hypermedia systems. *SIGWEB Newsletter*, Vol: 8, Num 3.
- [Hamadeh, 1993] Hamadeh, A. (1993). *BOCAL ou la Boîte comme Paradigme programmatique : Application a la Simulation de Systèmes Productifs*. Thèse de doctorat de l'université Joseph Fourier - Grenoble I, Laboratoire ARTEMIS, 27 Octobre. 136 pages.
- [Hanser, 2003] Hanser, D. (2003). *Proposition d'un modèle d'auto coordination en situation de conception, application au domaine du bâtiment*. Thèse de Doctorat de l'Institut National Polytechnique de Lorraine Discipline : Sciences de l'Architecture, 31 Octobre. 215 pages.
- [Hamadeh, 1993] Hamadeh, A. (1993). *Bocal ou la Boîte comme paradigme programmatique : Application a la simulation des systèmes productifs*. Thèse de doctorat de l'université de Joseph Fourier Grenoble I, 27 Octobre. 137 pages.
- [Hoogdtoel, 1996] Hoogdtoel, F. (1996). *Une approche organisationnelle du travail coopératif assisté par ordinateur Application au projet Co-Learn*. Thèse de doctorat de l'université de sciences et technologies de LILLE. 289 pages.
- [Howell et McNab, 1998a] Howell, F.W. and McNab R. (1998). simjava: A discrete event simulation library for Java. In: *Proceedings of International Conference on Web-Based Modeling and Simulation*. International Society for Computer Simulation January.

- [Howell et McNab, 1998b] Howell, F.W. and McNab R. (1998). simjava: a discrete event simulation package for Java with applications in computer systems modeling. In : *First International Conference on Web-based Modelling and Simulation*, San Diego CA, Society for Computer Simulation, January .
- [Horstmann and Bentley, 1997] Horstmann, T., Bentley R. (1997). Distributed Authoring on the Web with the BSCW Shared Workspace System. *Standardview*, Vol. 5, Num 1, March 1997. pp. 9-16.
- [Iazeolla et D'Ambrogio, 1998] Iazeolla, G. and D'Ambrogio A. (1998) A web-based environment for the reuse of simulation models. In *1998 International Conference on Web-Based Modeling and Simulation*. Society for Computer Simulation International (SCS), January.
- [Jense et al., 1997] Jense, G.J., Kuijpers N.H.L. and Dumay A.C.M.. (1997). DIS and HLA: Connecting people, Simulations and Simulators in the Military, Space and Civil Domains. *48th International Astronautical Congress*, Turin, Italy, October 6-10.
- [John et al., 2004] John A. Miller, Gregory Baramidze, Paul A. Fishwick and Amit P. Sheth. (2004). Investigating Ontologies for Simulation Modeling. *Proceedings of the 37th Annual Simulation Symposium (ANSS'04)*, Arlington, Virginia April. pp. 55-71.
- [Kapuno and Nagarur, 1999] Kapuno Jr., R. R. and N. N. Nagarur. (1999). SimProd: A Web-Based Flexible Simulation Package for Production Systems. *Electronic Journal of the School of Advanced Technologies*, Asian Institute of Technologies, Vol.1, No.2. ISSN: 1513-1432.
- [Kim et Fishwick, 1998] Kim, Y., Fishwick P. A.. (1998). The Design of Human Computer Interface for Multimodeling Object Oriented Simulation Environment. *SPIE Aerosense Conference*, Orlando, Florida , April.
- [Klein et al., 1998] Klein, U., Straburger S. and Beikrich J. (1998). Distributed simulation with JavaGPSS based on the high level architecture. In *Proceedings of the 1998 SCS International Conference on Web-Based Modeling and Simulation*, P. A. Fishwick, D. R. C. Hill, and R. Smith, eds. San Diego, CA, 11-14 January. pp. 85 - 90.
- [KLC, 2007] KLC (Dictionnaire de la maîtrise d'ouvrage des projets informatiques) (2007). , Site Web accessible à l'url: http://www.klc.fr/FR/METHO/DicoSI_IE_FR.html
- [Knave et Stackenland, 2003] Knave, E., Stackenland E. (2003). *Remote Collaborative Environment to Support Simulation Development*. MSc Thesis, Royal Institute of Technology (KTH),

- Stockholm Sweden, in collaboration with National University of Singapore (NUS), February.
- [Lang et al, 1995] Lang, U., Peltier J.P., Christ P., Rill S. , Rantzau D., Nebel H., Wierse A., Lang R., Causse S., Juaneda F., Grave M., and Haas P. (1995). COVISE: Perspectives of Collaborative Supercomputing and Networking in European Aerospace Research and Industry, Future Generation Computer Systems (FGCS) Elsevier Science, Vol 11, Num 4-5, pp. 419-430.
- [Laurillau, 2002] Laurillau, Y. (2002). *Conception et réalisation logicielles pour les collecticiels centrées sur l'activité de groupe : le modèle et la plate-forme Clover*. Thèse de doctorat de l'université Joseph Fourier - Grenoble I, Laboratoire de Communication Langagière et Interaction Personne - Système Fédération IMAG, 10 Septembre. 216 pages.
- [Legedza, 1995] Legedza, U. (1995). *Reducing Synchronisation Overhead in Parallel Simulation*. Master's Thesis of Massachusetts Institute of Technology, June.
- [Livet, 2002] Livet, A. (2002). *Modélisation des systèmes physiques de production pour l'évaluation des coûts des*. Thèse de doctorat de l'université de Louis Pasteur Strasbourg, 10 Décembre juillet. 133 pages.
- [Lukasz et al., 1997] Lukasz Beca, Gang Cheng, Geoffrey Fox, Tomasz Jurga, Konrad Olszewski, Marek Podgorny, Piotr Sokolowski, and Krzysztof Walczak.(1997). Web Technologies for Collaborative Visualization and Simulation. *NPAC Technical Report SCCS-786*, January.
- [Letouzey, 2001] Letouzey, A. (2001). *Ordonnancement interactif basé sur des indicateurs : Applications à la gestion de commandes incertaines et à l'affectation des opérateurs*. Doctorat de l'Institut National Polytechnique de Toulouse, Equipe Production Automatisée du Laboratoire Génie de Production de l'Ecole Nationale d'Ingénieurs de Tarbes, 19 décembre. Pages 233.
- [Mackulak et al., 1994] Mackulak, G. T. , Cochran J.K., Savory P. A. (1994). Ascertaining Important Features for Industrial Simulation Environments. *Simulation* Vol. 63, Num 4, October. p. 211-221
- [Malcurat, 2001] Malcurat, O. (2001). *Spécification d'un environnement logiciel d'assistance au travail coopératif dans le secteur de l'architecture et du BTP*. Thèse de doctorat, Sciences pour l'architecture. Institut National Polytechnique de Lorraine. 153 pages.

- [McNab et Howell, 1996] McNab, R. and Howell F.W. (1996). Using Java for Discrete Event Simulation. In *proc. Twelfth UK Computer and Telecommunications Performance Engineering Workshop (UKPEW)*, University of Edinburgh, 219-228.
- [Mann et Waldmann, 1998] Mann, H. and Waldmann L. (1998). Web-based modelling and simulation of multidisciplinary engineering systems. In *P. Fishwick, D. Hill, and R. Smith, editors, The International Conference on Web-Based Modelling and Simulation*. San Diego, California, January. The Society for Computer Simulation International. pages 183 - 8.
- [Navarre, 2001] Navarre, D. (2001). *Contribution à l'ingénierie en Interaction Homme-Machine : Une technique de description formelle et un environnement pour une modélisation et une exploitation synergiques des tâches et du système*. Thèse de Doctorat, Université de Toulouse I, 2 juillet. 218 pages.
- [Owezarski, 1996] Owezarski, P. (1996). *Conception et formalisation d'une application de visioconférence coopérative. Application et extension pour la téléformation*. Thèse de doctorat de l'université Paul Sabatier de Toulouse III (Sciences). 154 pages.
- [Page et Opper, 2000] Page E.H., Opper J.M. (2000). Investigating the Application of Web-Based Simulation Principles within the Architecture for a Next-Generation Computer Generated Forces Model. *Future Generation Computer Systems*, Vol 17 Num 2, 159 – 169.
- [Page et al., 1999] Page, E.H., Buss, A., Fishwick, P.A., Healy, K.J., Nance, R.E. and Paul, R.J. (1999). Web-Based Simulation: Revolution or Evolution? Submitted to: *ACM Transactions on Modeling and Computer Simulation*, February.
- [Page et al., 1998a] Page, E.H., Buss, A., Fishwick, P.A., Healy, K.J., Nance, R.E. and Paul, R.J. (1998). The Modeling Methodological Impacts of Web-Based Simulation. In: *Proceedings of the 1998 SCS International Conference on Web-Based Modeling and Simulation*. San Diego, CA, 11-14 January. pp. 123-128.
- [Page, 1998b] Page, E.H. (1998). The Rise of Web-Based Simulation: Implications for the High Level Architecture. In: *Proceedings of the 1998 Winter Simulation Conference*. Washington, DC, 13-16 December. pp. 1663-1668.
- [Page et al., 1998c] Page, E.H, S. Griffin, and S. Rother. (1998). Providing Conceptual Framework Support for Distributed WebBased Simulation within the High Level Architecture. In *Proceedings of SPIE: Enabling Technologies for Simulation Science II*, 287-292.

- [Payet, 2003] Payet, D. (2003). *L'enrichissement de message comme support pour la composition logicielle*. Thèse de doctorat de l'université de Montpellier II, 04 Décembre. 228 pages.
- [Peter Lorenz et al., 1998] Peter Lorenz, Thomas J. Schriber, Daniel Beier, Stefan Hiller: (1998). Web Support for a Simulation and Animation Course. *SimVis 1998*: 67-79.
- [Peter Lorenz et al., 1997] Peter Lorenz, Heiko Dorwarth, Klaus-Christoph Ritter, Thomas J. Schriber: (1997). Towards a Web Based Simulation Environment. *Winter Simulation Conference 1997*. pp 1338-1344.
- [Peter et al., 1997] Peter J. Spellman , Jane N. Mosier , Lucy M. Deus , Jay A. Carlson. (1997). Collaborative virtual workspace. *Proceedings of the international ACM SIGGROUP conference on Supporting group work : the integration challenge: the integration challenge*, November 16-19, 1997, Phoenix, Arizona, United States. p.197-203
- [Pidd et al., 1999] Pidd, M., Oses, N., and Cassel, R. A.. (1999) Component-based simulation on the web? In *Proceedings of the 1999 Winter Simulation Conference (WSC'99)*, December.
- [Pidd et Cassel, 2000] Pidd M. & Cassel R.A. (2000) Web based simulation using Java. *IEEE Potentials*, Vol. 19 Num 1, 11-16.
- [Roseman et Greenberg, 1997] Roseman M. and Greenberg S. (1997). Simplifying component development in an integrated groupware environment. In *Proceedings of the ACM UIST'97 Symposium on User Interface Software and Technology*, Banff, Alberta, ACM Press. pp 65-72
- [Salber, 1995] Salber, D. (1995). *De l'interaction homme-machine individuelle aux systèmes multi-utilisateurs, L'exemple de la communication homme-homme médiatisée*. Thèse de doctorat de l'Université Joseph Fourier - Grenoble I, 8 septembre. 302 pages.
- [Salvador et al., 1996] Salvador, T., Scholtz, J., Larson, J. (1996) The Denver Model for Groupware Design. *ACM SIGCHI Bulletin*. Vol. 28. pp 52--58.
- [Seibt et al., 1998] Seibt, F., Schumann M., Beikirch J. (1998). Concept and Components for a Web-based Simulation Environment (WBSE). *International Conference on Web-Based Modeling & Simulation*. San Diego CA, 1998.
- [Siegel, 1999] Siegel, G. (1999). *PROSIT : Un Environnement pour la programmation de simulation à événements discrets*. Thèse de doctorat de l'université de Nice-Sophia Antipolis, URF-Sciences, 29 septembre. 190 pages.

- [Sire, 2000] Sire, Stéphane. (2000). *La Collaboration Directe : un paradigme d'interaction pour le travail collaboratif assisté par ordinateur*. Thèse de doctorat de l'université de Toulouse I, 1 février. 198 pages.
- [Sire et Chatty, 1999] Sire, S., Chatty S. (1999). Les collecticiels peuvent-ils préserver nos compétences sociales de collaboration ? In *Revue Informations In Cognito n°13/1999*, pp. 23-36.
- [Sire et Chatty, 1998] Sire, S., Chatty S. (1998). Vers des interfaces à collaboration directe pour le travail de groupe. In *Communication, société et Internet, actes du colloque GRESICO98*. 10-11 septembre, Vannes, éditions de l'Harmattan, pp. 259-275.
- [Sire, 1997] Sire, S. (1997). Un cadre à la conception d'interfaces pour le collecticiel : la collaboration directe. In *neuvièmes journées sur l'ingénierie de l'Interaction Homme-Machine, IHM'97*. 10 - 12 septembre, Futuroscope, éditions Cépaduès, Toulouse, pp. 143-149.
- [Shen, 1998] Shen, C.-C. (1998). Discrete-Event Simulation on the Internet and the Web. In *Proceedings of the 1998 SCS International Conference on Web-Based Modeling and Simulation*, San Diego, CA, 11-14 January. pp. 57-62.
- [Tarpin-Bernard, 1997a] Tarpin-Bernard, F. (1997). Apports des Collecticiels à l'ingénierie. *PRIMECA'97*. La Plagne. Vol. 3. No. 2. Avril. p. 333-338
- [Tarpin-Bernard, 1997b] Tarpin-Bernard, F. (1997). *Travail Coopératif Synchronisé Assisté par Ordinateur Approche AMF-C*. Thèse de doctorat de l'université de Lyon, 1 juillet. 145 pages.
- [Tarpin-Bernard, 1996] Tarpin-Bernard, F., Bertrand, D. (1996) Ergonomie du Travail Coopératif en Conception. *ERGO-IA'96*, Biarritz, Octobre.
- [Tarpin-Bernard et al., 1995] Tarpin-Bernard, F., Bertrand D., Ouadou K. (1995) Collecticiel : Modèle et Architecture et validation sur un exemple. *4° journées internationales Mondes Réels et Virtuels*. Montpellier, Juin. p. 401-414.
- [Taylor, 2000] Taylor, S.J.E. (2000). NetMeeting: a tool for collaborative simulation modeling *I. J. of Simulation Systems, Science & Technology*, Vol. 1 No 1-2:59-68.
- [Tewoldeberhan et al., 2002] Tewoldeberhan, T. W., Verbraeck A., Valentin E., Bardonnnet G. (2002). Software evaluation and selection: an evaluation and selection methodology for discrete-event simulation software. *Proceedings of the 34th Winter Simulation Conference: Exploring New Frontiers*, San Diego, California, USA, December 8-11. pp: 67-75

- [Trevor et al., 1997] Trevor, J., Koch T., Woetzel G. (1997). MetaWeb: Bringing Synchronous Groupware on the World Wide Web. *European Conference on Computer Supported Cooperative Work (ECSCW'97)*, Lancaster (UK).
- [Vacherand-Revel et Tarpin-Bernard, 1997] Vacherand-Revel J., Tarpin-Bernard F. (1997). Méthodologie d'évaluation de l'usage d'un collecticiel. *9^{ème} journées sur l'ingénierie des Interfaces homme-machine (IHM'97)*. Poitiers, France, 11-13 sept. pp. 71-78.
- [Van der Veer et al., 1996] Van der Veer, G.C., Lenting B.F., Bergevoet, B.A.J.(1996). GTA: Groupware Task Analysis - Modeling Complexity. *Acta Psychologica*, Vol. 91, pp. 297-322.
- [Vapillon, 2000] Vapillon, J. (2000). *Contribution à l'étude de la conversation dans le cadre du travail coopératif assisté par ordinateur (TCAO)*. Thèse de doctorat de l'université de Paris Sud, UFR Scientifique, 24 Octobre. 302 pages.
- [Wilcox et Pooley, 1998] Wilcox, P.A. and Pooley R.J. (1998). Factory Simulation across the World Wide Web using Java. *Proceedings of the 1998 International Conference on Web-based Modeling and Simulation*, SCS Western Multiconference, San Diego, January 11-14. pp 174-179.
- [Wladimir et al., 2004] Wladimir, A. F., Hirata C. M., Edgar T. Y. (2004). GroupSim: A Collaborative Environment for Discrete Event Simulation Software Development for the World Wide Web. *SagePub*, Vol. 80, Num 6. pp 257-272.

Sites WWW

- [GPA662, 2007] Cours GPA662 : Modélisation et simulation de systèmes de production
<http://www.gpa.etsmtl.ca/cours/gpa662/Cours1/>.

Annexes

Résumé :

Mes travaux de recherches se situent au croisement de deux domaines : le travail coopératif assisté par ordinateur (TCAO) et la Simulation. C'est une contribution à l'ingénierie des outils de simulation coopérative. Nous avons adopté un regard nouveau sur les outils de simulation existants, en étudiant comment de tels outils peuvent intégrer les concepts du TCAO et des aspects de la conception comme ceux relevant de l'ergonomie ou de l'activité de groupe. Ainsi nous avons fixé comme objectif la mise en œuvre d'un collecticiel de simulation en prenant en compte des aspects sociaux, éthiques et ergonomiques. La contribution dans ce travail de recherche est essentiellement la proposition des modèles d'architecture conceptuelle et logicielle d'un collecticiel de simulation. Nous avons orienté nos recherches vers la définition d'un modèle de projet de simulation coopératif, puis vers la spécification de solutions informatiques permettant d'assister les membres d'un projet de simulation dans leurs activités coopératives. Nos recherches ont permis, d'une part, de rationaliser la définition d'un collecticiel appliqué au domaine de la simulation en nous rapprochant d'une typologie de modélisation standard, et d'autre part, de proposer une solution technique basée sur les offres actuelles du domaine du TCAO.

Mots Clés : Simulation, modélisation, TCAO, Simulation Coopérative, Collecticiel, Interaction Homme-Machine

Abstract:

My research works are set in the crossing of two fields: the CSCW (Computer Supported Cooperative Work) and the Simulation. It is a contribution to the cooperative simulation tools engineering. We have adopted a new glance on the existing tools of simulation, by studying how such tools can integrate the concepts of the CSCW and the aspects of the conception like those relative to ergonomics or to the group activity. So we fixed as objective the implementation of a simulation groupware taking into account social, ethical and ergonomic aspects. The contribution in this research work is essentially the proposition of the models of conceptual and software architecture of a groupware of simulation. Our researches were turned towards to the definition of a model of cooperative simulation project, then to the specification of computer solutions allowing to assist the members of a simulation project in their cooperative activities. Our researches allowed, on the one hand, to rationalize the definition of a groupware applied to the simulation field by moving closer to a standard modeling typology, and on the other hand, to propose a technical solution based on the current offers of the CSCW.

Keywords: Simulation, modeling, CSCW, Cooperative Simulation, Groupware, Human-Computer Interaction.