

THESE

présentée en vue de l'obtention du
DOCTORAT de l'Université
des Sciences et Technologies de Lille
Spécialité: **INFORMATIQUE**

par

Fouad YOUSFI

Soutenue le 22 mars 1996

***PLACO: Modélisation par Workflow et Conception
d'un Système de Planification Coopérative
Application aux unités de soins***



Jury:

- Président:** M. Jean-Luc DEKEYSER (Université de Lille I)
- Directeurs:** M. Jean-Marc GEIB (Université de Lille I),
M. Régis BEUSCART (Université de Lille II)
- Rapporteurs:** M. Jacques DEMONGEOT (Université de Grenoble),
Mme. Anne NICOLLE (Université de Caen)
- Examineur:** M. Michel DEMEESTER (Premium, S.A, Belgique)

REMERCIEMENTS

Je remercie sincèrement Jacques Demongeot, professeur à l'INPG de Grenoble, et Anne Nicolle, professeur à l'université de Caen, qui m'ont fait l'honneur d'être rapporteurs de ce travail de thèse.

Je veux remercier Jean-Luc Dekeyser d'avoir accepté de présider le jury de cette thèse, ainsi que Michel Demeester d'avoir accepté d'examiner cette dernière.

Je tiens à remercier également Régis Beuscart, directeur du CERIM, ainsi que Jean-Marc Geib, directeur du LIFL, pour m'avoir proposé un sujet très enrichissant qui s'inscrit au sein du programme du groupe inter-laboratoire GANYMEDE, pour l'aide précieuse qu'ils m'ont apportée le long de cette thèse, ainsi que pour les critiques constructives ayant servi à la bonne rédaction de ce mémoire.

Je tiens aussi à remercier Bertrand Bossard, Eric Dufresne, Nathalie Souf, Jean-Marie Renard et Alain Duhamel pour leur aide au niveau professionnel et amical, ainsi que pour les conseils qu'ils m'ont prodigués.

Mes remerciements s'adressent également à Valérie qui a su être près de moi quand j'avais besoin d'elle, et ainsi qu'à Taoufik, pour le temps qu'ils ont passé à relire ma thèse.

J'ai une pensée toute particulière envers tous les autres membres du CERIM pour leur gentillesse et leur disponibilité.

Enfin, je remercie toutes les personnes, qui de près ou de loin, ont su m'aider, ne serait ce que par un sourire, à surmonter les problèmes et les découragements qu'induit tout travail de recherche.

A mes parents,

Table des Matières

Liste des Figures	v
Introduction	1
Plan général de la thèse	9

PARTIE I

CSCW et Informatique Médicale

CHAPITRE 1

Travail Coopératif, CSCW et Groupware	11
1.1 Introduction	11
1.2 Le CSCW et le Travail Coopératif	12
1.2.1 Définition	12
1.2.2 Les premiers pas du CSCW	12
1.2.3 Le groupe et le CSCW	13
1.2.4 Les concepts-clés du CSCW	14
1.2.5 Les formes de coopération	16
1.2.6 Les espaces de la coopération	17
1.2.7 Les systèmes distribués et le CSCW	20
1.3 Le Groupware	22
1.3.1 Définition du groupware	22
1.3.2 Typologies des systèmes groupware	23
1.4 Conclusion	27

CHAPITRE 2	Le Workflow	29
2.1	Introduction.	29
2.2	La Gestion de Workflow	29
2.3	La perspective du langage/action	30
2.3.1	Analyse conversationnelle	32
2.4	Théorie de la coordination	33
2.4.1	Qu'est ce que la coordination?	33
2.4.2	Autres définitions	34
2.4.3	Composants	35
2.4.4	Interdépendance	36
2.4.5	Processus sous-jacents aux situations de coordination	37
2.5	L'approche AWA (Action Workflow Approach)	38
2.5.1	Perspective	38
2.5.2	Processus de travail	38
2.5.3	Comparaison avec les autres approches de gestion workflow	40
2.5.4	Les inconvénients de l'approche AWA	41
2.6	Outils de Workflow	42
2.6.1	Approche DIGITAL	42
2.6.2	Approche LotusNotes	46
2.7	Conclusion	48
CHAPITRE 3	Informatique Médicale.	50
3.1	Introduction.	50
3.2	Gestion de l'information en médecine	51
3.2.1	Les systèmes d'information hospitaliers (SIH)	51
3.2.2	L'intégration	54
3.2.3	La coopération	56
3.3	L'USI et le «Tableau Blanc» dans la Chambre du Patient	57
3.3.1	Définition d'une unité de soin	58
3.3.2	Fonctionnement d'une unité de soin	58
3.3.3	Spécificités des USI	60
3.3.4	L'information supportée par le tableau blanc en USI	61
3.3.5	La solution des «post-it»	62
3.4	Conclusion	64

PARTIE II	La Proposition PLACO	
CHAPITRE 4	Coopération et Coordination: Application à l'USI	67
	4.1 Analyse Socio-Psychologique des «breakdowns»	67
	4.1.1 Définition	67
	4.1.2 Hypothèses de l'étude	68
	4.1.3 Méthodologie de l'étude	68
	4.1.4 Analyse des résultats	69
	4.1.5 Conclusion de cette analyse	73
	4.2 Choix du thème d'application	74
	4.3 Proposition de la solution PLACO: une Architecture Coopérative Distribuée	75
	4.3.1 Spécifications	76
	4.3.2 Conclusion	79
CHAPITRE 5	Le Modèle Workflow Asynchrone de PLACO appliqué aux USI	83
	5.1 Introduction	83
	Définition	86
	5.2 Le modèle Abstrait de PLACO	86
	5.2.1 Le modèle Asynchrone	86
	5.2.2 Sous-Cycle de Délégation	94
	5.2.3 Sous-cycle de Clarification	95
	5.2.4 Propriétés des liens des sous-cycles de Délégation et de Clarification	96
	5.2.5 Généralisation du modèle	99
	5.2.6 Exemple d'un cas pratique de planification coopérative	102
	5.2.7 Conclusion	104
	5.3 Le Modèle WF d'exécution de PLACO	105
	5.3.1 Message semi-structuré	106
	5.3.2 Le modèle Atomique avec les messages	108
	5.3.3 Circuit des messages en situations de délégation et de clarification	111
	5.3.4 Généralisation des messages à l'ensemble du Modèle Abstrait	114
	5.3.5 Exemple d'un cas réel: instanciation du modèle d'exécution avec les messages	115
	5.3.6 Conclusion	118
CHAPITRE 6	Le Tableau Blanc et La Métaphore du «Post-it»	120
	6.1 Introduction	120

6.2	La Solution pour l'Informatisation du «Tableau Blanc»	121
6.2.1	L'information manipulée	121
6.2.2	L'architecture proposée	128
6.2.3	Le «Post-it»: version informatique	129
6.2.4	Exemple de scénarios avec gestion de «post-it»:	135
6.3	Conclusion	140
CHAPITRE 7	Réalisation et Evaluation du Prototype de PLACO	141
7.1	La Représentation Orientée Objets de la Solution PLACO	142
7.1.1	Pourquoi l'approche orientée objets?	142
7.1.2	La représentation orientée objets de PLACO	144
7.2	Réalisation du Prototype de PLACO	150
7.2.1	Le langage Smalltalk	150
7.2.2	Description du prototype de PLACO	152
7.2.3	Aspects techniques du prototype	157
7.3	Evaluation du Prototype PLACO	162
7.3.1	Le modèle PLACO	162
7.3.2	Evaluation des fonctions du prototype PLACO	163
7.4	Perspectives	170
7.4.1	La distribution de l'application	170
7.4.2	Les interfaces homme-machine	170
	 Conclusion et Perspectives	 172
	 Références Bibliographiques	 178
	 Graphe d'Héritage de la Solution PLACO	 A.1
	 Exemples d'Interfaces du Prototype	 B.3
	 Spécimen de Feuilles de Prescription et de Surveillance en Réanimation Plyvalente	 C.4

Liste des Figures

FIGURE 1	La nature géographique dans les systèmes CSCW.	18
FIGURE 2	Classes des systèmes CSCW supportant la coopération synchrone et asynchrone.	19
FIGURE 3	Taxonomie des activités de groupe suivant les dimensions de l'espace et du temps.	26
FIGURE 4	La conversation pour action. Ce graphe d'état-transition représente la conversation pour action. Elle est amorcée par une demande de la part du locuteur A vers le locuteur B. Les ronds représentent les états de la conversation. Les flèches étiquetées représentent les actes de langage sous-jacents. Les ronds en gras représentent les états qui marquent l'achèvement de la conversation.	32
FIGURE 5	Les mécanismes de coordination d'après MINTZBERG.	36
FIGURE 6	La boucle de workflow-action d'après ATI.	39
FIGURE 7	Vue globale du modèle des systèmes de workflow selon l'approche DWA (Digital Inc.) [Digi93].	44

FIGURE 8	Le concept du système TeamRoute (Digital Inc.) [Digi93]. 45
FIGURE 9	Principe de fonctionnement des systèmes conçus selon les principes de «gestion de la connaissance» [Lotu93]..... 47
FIGURE 10	Les composants de l'architecture Notes [Lotu93]. 48
FIGURE 11	Exemple d'une architecture de SIH distribuée basée sur des serveurs. 54
FIGURE 12	L'approche d'intégration ISAR [ISAR94b]. 56
FIGURE 13	L'architecture logicielle distribuée de PLACO. Elle se compose essentiellement de trois tableaux qui sont: le tableau A contenant les objets d'information, le tableau B spécialisé dans le contrôle de cohérence des objets et le tableau C de contrôle et de gestion des interactions entre acteurs. Ces derniers, au moyen d'un processus de conversation, interagissent en modifiant et en créant des objets d'information sur le tableau A. 73
FIGURE 14	Le cycle élémentaire workflow appliqué aux unités de soins intensifs. Ce modèle identifie les actions fondamentales d'une activité de soin..... 82
FIGURE 15	Le cycle principal de WF appliqué aux unités de soins intensifs avec le sous-cycle de délégation attaché à la phase de prescription. Le cycle 1.1 s'appelle un sous-cycle ou un cycle secondaire par rapport au cycle principal 0.0..... 88
FIGURE 16	Le cycle principal de WF appliqué aux unités de soins intensifs avec le sous-cycle de clarification attaché à la phase de prescription. Le cycle 1.1 s'appelle un sous-cycle ou un cycle secondaire par rapport au cycle principal 0.0..... 89

-
-
- FIGURE 17** Le cycle workflow principal appliqué aux unités de soins intensifs avec les sous-cycles de délégation et de clarification. Ce modèle identifie les actions fondamentales d'une activité de soin. L'acteur est capable d'exécuter la tâche de prescription s'il dispose de toute l'information nécessaire à l'accomplissement de la tâche, de déléguer sa tâche à un autre acteur, et/ou de demander une clarification afin de disposer de l'information indispensable à la réalisation d'une prescription. Il peut également générer un arrêt de prescription (faire une action nulle) si l'information obtenue à l'issue de ces deux transactions est insuffisante.
..... 91
- FIGURE 18** Les sous-cycles de délégation et de clarification généralisés à toutes les phases du cycle principal 0.0. Les liens dynamiques et conditionnels reliant ces sous-cycles mettent en évidence la possibilité, à tout moment, pour un acteur de déléguer sa tâche, demander ou fournir une clarification concernant son rôle. La conversation de type délégation s'établit avec un acteur habilité à remplir le même rôle que l'acteur demandeur. La clarification se passe avec l'acteur responsable de la tâche précédente dans le cycle principal, ou avec celui qui, potentiellement, sera le responsable de la tâche suivante.
..... 95
- FIGURE 19** Instanciation du cycle principal de workflow pour une prescription en situation normale. 98
- FIGURE 20** Exemple de message semi-structuré. Il s'agit d'un message de type Annonce_Réunion, envoyé par «Beuscart» à l'intention de l'équipe PLACO pour leur annoncer la date et le lieu de la prochaine réunion PLACO. Les huit premiers champs contiennent des informations structurées pouvant être analysées facilement. Quant au dernier champ, il

- contient du texte libre, donc de l'information non-structurée..... 101
- FIGURE 21 Exemple de message semi-structuré. Il s'agit d'un message de type Clarification-Prscription, envoyé par «Médecin_X» à l'intention de l'infirmière-chef «Infirmière_Y» chargée de la planification du traitement. Ce message concerne le patient «Patient_X» et la planification du traitement prescrit «Prscription_P» dans la phase «Phase_F». Le but du message est de fournir des clarifications contenues dans le champ Clarifications qui contient du texte libre (non-structuré). 102
- FIGURE 22 Le modèle WF fonctionnel de PLACO avec la prise en charge de la coordination par des messages. Il s'agit du cycle de WF principal étendu (adapté aux unités de soins intensifs). Dans chaque phase, un message M_i (R_j) est généré par le système pour un rôle R_j . M_1 (R_2) sert à spécifier dans quelles conditions l'action $Ac(R_1)$ (la prescription) a été réalisée et annonce la réalisation de l'action $Ac(R_2)$ qui est la planification. M_2 (R_3) annonce la fin de l'action $Ac(R_2)$ et contient l'information concernant la manière dont il faut exécuter l'action $Ac(R_3)$ qui est la dispensation. M_3 (R_4) indique la façon dont l'action $Ac(R_3)$ a été accomplie (i.e. la prescription a été administrée effectivement au patient) et amorce l'action $Ac(R_4)$. M_4 (R_1) annonce la fin du cycle de workflow. Il indique la décision obtenue quant à la suite..... 104
- FIGURE 23 Circuit des messages générés lors de la phase de prescription dans le workflow principal. Les cycles secondaires de délégation et de clarification induisent des messages supplémentaires qui viennent s'ajouter aux messages principaux produits dans le cycle principal. L'ensemble de ces messages crée

	un flux permettant de tracer l'évolution de l'information.	107
FIGURE 24	Le circuit des messages générés lors des quatre phases du cycle principal. Les messages produits dans les cycles secondaires de délégation et de clarification, pour chaque phase, créent un flux d'information indispensable pour tracer l'évolution du workflow dans le cycle principal.	109
FIGURE 25	Instanciation du cycle principal de workflow pour une prescription de Dobutrex en situation normale.	111
FIGURE 26	Graphe d'état-transition d'un objet prescrit dans le cycle workflow. A l'état initial, il est vide. Les trois états qui suivent sont transitoires et correspondent aux phase de prescription, de planification et de dispensation. L'état final correspond à l'objet évalué.	126
FIGURE 27	Fonctionnement des agents de la solution PLACO. L'acteur X interagit avec son IUG. Celle-ci communique avec l'agent d'interface qui prend en charge les interactions de X et les transmet sous forme de requêtes à l'agent de workflow. Ce dernier, par notification, communique à l'agent de «post-it» la nature des événements survenus dans l'espace d'information partagé afin qu'il puisse mettre à jour les «post-it» existants et, éventuellement, en créer d'autres qu'il attachera aux acteurs destinataires et à l'objet D1.	130
FIGURE 28	L'architecture que nous proposons est implantée en couches. Les agents d'interface sont représentés dans le système par leur interface spécifique.....	131
FIGURE 29	Un exemple de «post-it» créé dans la phase de prescription. Il contient des informations destinées à un groupe d'infirmières «groupe_I1». Il fait éga-	

- lement référence à plusieurs objets tels que la prescription «Prescription_P1» et l'action «action_A1». Les liens de type «lié à» sont actifs et permettent aux destinataires un accès direct aux objets dits référencés..... 133
- FIGURE 30 Un exemple d'envoi de «post-it» de type message de coordination attaché à un groupe d'acteurs. Le «post-it» P est envoyé par un acteur_1 vers le groupe G1 et référence plusieurs objets tels que la prescription D1, le patient P, etc. 135
- FIGURE 31 Un exemple d'envoi de «post-it» de type message libre. Le «post-it» P, envoyé par un acteur_1 vers un acteur_2, contient des informations non structurées. 136
- FIGURE 32 Un exemple d'un scénario de prescription d'un traitement D1: Dobutrex en SAP. Un «post-it» workflow P1 créé et attaché à la nouvelle prescription avec un «post-it» de coordination R1 envoyé aux destinataires susceptibles d'exécuter la future action, qu'est la planification, sur cet objet. 138
- FIGURE 33 Un exemple d'un scénario de planification d'un objet prescrit D1. Un «post-it» workflow P2 est créé et est attaché à D1 avec un «post-it» de coordination R2 envoyé aux acteurs susceptibles d'exécuter la future action sur cet objet. Avant la création de P2 et de R2, le gestionnaire de «post-it» détruit les anciens «post-it» qui sont considérés obsolètes (P1 et R1) étant donné que l'action de planification qui leur est associée a été effectivement réalisée..... 139
- FIGURE 34 Un exemple d'un scénario de dispensation d'un objet prescrit D1. Un «post-it» workflow P3 est créé et est attaché à D1 avec un «post-it» de coordination R3 envoyé aux acteurs susceptibles d'exécuter la future action sur cet objet. Avant la création de

	P3 et de R3, le gestionnaire de «post-it» détruit les anciens «post-it» qui sont considérés obsolètes (P2 et R2) étant donné que l'action de dispensation qui leur est associée a été effectuée.....	140
FIGURE 35	Un exemple d'un scénario d'évaluation d'un objet prescrit D1. Un «post-it» workflow P4 est créé et est attaché à D1 avec un «post-it» de coordination R4 envoyé aux acteurs susceptibles d'exécuter la future action sur cet objet. Avant la création de P4 et de R4, le gestionnaire de «post-it» détruit les anciens «post-it» qui sont considérés obsolètes (P3 et R3) étant donné que l'action d'évaluation qui leur est associée a été effectuée.....	141
FIGURE 36	Diagramme des classes principales du système PLACO. Il se compose de quatre classes: la classe ObjetPersonne, la classe ObjetPrescrit, la classe WorkflowItem et la classe PostIt.....	150
FIGURE 37	Graphe d'héritage des classes d'objets de PLACO avec leurs liens de composition.....	153
FIGURE 38	Utilisation de la mémoire partagée pour faciliter la gestion des «post-it».....	158
FIGURE 39	Le modèle MVC de SmallTalk-80 correspondant au Modèle, à la Vue et au Contrôleur.....	163

Introduction

CSCW et Groupware

L'étude des systèmes dédiés à aider un travail commun à plusieurs personnes a donné naissance à une nouvelle discipline scientifique appelée CSCW: *Computer-Supported Cooperative Work*, acronyme dont la traduction en français donne le *Travail Coopératif Assisté par Ordinateur* ou encore, d'après l'AFCEP, *Activités Multi-Participantes Informatisées*.

Le CSCW est un domaine de recherche multi-disciplinaire relativement nouveau. Il vise à étudier le travail coopératif au sein des groupes, à définir des théories explicatives, et permet de concevoir des systèmes informatiques, qu'on appelle *Groupware* ou *Collecticiels*, supportant les activités de groupes dans les organisations [Gree91]. Cette discipline, récente de surcroît, est encore en développement. De nombreux problèmes théoriques inhérents à la nature sociale complexe du travail sont encore en suspens et méritent qu'on leur accorde une grande attention.

Comme l'a souligné ELLIS dans [Elli91], les collecticiels relèvent de cinq disciplines: *les systèmes distribués, la communication, l'interaction homme-machine, l'intelligence artificielle et la théorie sociale*. Les collecticiels combinent souvent deux ou plus de ces cinq approches. Les collecticiels présents sur le marché sont de plus en plus nombreux. ELLIS propose une classification de ces derniers suivant les fonctionnalités qu'ils offrent [Elli91]:

1. **Les systèmes de messagerie** tels que le Email, «Information LENS» [Mal86b], «IMAIL» [Hogg85], etc.

2. Les éditeurs multi-utilisateurs tels que ForComment[®] [Oppe88], «Collaborative Editing Systems» (CES) [Greif86], «Shared Book» [Lewi88], «Quilt» [Fish88] [Lelan88], «Mercury» [Kais87], etc.
3. Les systèmes de co-décision ou les «Group Decision Support Systems» (GDSSs) tels que PlexCenter [App186].
4. Les systèmes de conférence: tels que RTCAL [Sari85], COGNOTER [Stef87].
5. Les systèmes de coordination: tels que le système COORDINATOR[®] [Flor88], Lotus-Notes[®] [Lotu93], LinkWorks[®] [Digit93].

L'application de la technologie CSCW au secteur de la santé est axée principalement autour de la télé-médecine et la télé-assistance médicale, laissant de coté les nombreux problèmes liés à la coordination des activités de soins. On pourrait s'étendre davantage sur les problèmes, de surcroît très divers, des applications coopératives. Mais notre intention, dans cette introduction, est de présenter cette nouvelle notion qu'est le travail coopératif et de mesurer sa complexité dans les environnements intégrant des activités de soins, afin de mieux appréhender notre motivation dans ce projet.

Le travail coopératif en unités médicales

L'analyse du travail dans une unité médicale permet de démontrer aisément la nature coopérative des différentes tâches réalisées. Très souvent, l'analyse fonctionnelle, en particulier des activités médicales, s'est faite de façon descendante. Elle a permis d'identifier les tâches effectuées, les rôles du médecin, de l'infirmière, de l'aide-soignante et de l'hôtesse, de manière séparée et indépendante. Cette perspective a des répercussions sur la conception et la mise au point des outils informatiques qui leur étaient destinés. En effet, les notions de personnalisation et d'adaptation au contexte de travail ont favorisé la séparation des activités de chacun en leur donnant un accès personnalisé aux informations et une vision restrictive de l'ensemble du processus de travail.

De ce fait, la prise en charge efficace du malade dépend de la bonne coordination des différents soignants, ce qui concourt à la qualité de l'organisation du travail et par conséquent, à la qualité des soins. Dans [Hodg92], l'auteur démontre que la majorité des

erreurs de fonctionnement lors de la prise en charge est due aux erreurs dans la transmission d'information et à l'insuffisance des échanges inter-personnels.

Coordination

Le problème de coordination concerne l'intégration et l'ajustement harmonieux des efforts individuels vers l'accomplissement d'un but global [Elli91]. Les systèmes de coordination abordent ce problème de différentes façons. Concrètement, ils permettent aux utilisateurs de «voir» leurs actions ainsi que celles des autres s'accomplir dans un contexte de travail partagé. Les systèmes de coordination se caractérisent par les modèles sur lesquels ils sont basés. ELLIS en dénombre quatre types [Elli91]:

1. *modèles orientés formulaire* [Loch88].
2. *modèles orientés procédures* [Blaz89].
3. *modèles orientés conversation*.
4. *modèles orientés communication*.

Expérimentation

Afin de mettre en évidence l'importance de la coordination des tâches lors d'une prise en charge médicale, nous nous sommes intéressés aux modèles régissant le flux de travail dans les organisations et en particulier, dans les unités de soins intensifs et de réanimation. Nous avons mis en place une expérience afin d'analyser, sur le plan psycho-sociologique, les «breakdown», c'est-à-dire les erreurs, les anomalies, les incidents et les accidents qui surviennent quotidiennement dans les unités de soins intensifs (USI) [AIM-TANIT92].

Nous avons pu identifier plusieurs situations où la coopération et la coordination de tâches de l'ensemble du processus de soins échouent. Les résultats obtenus nous ont permis également de démontrer que la faible coordination est souvent à l'origine des «breakdowns» [AIM-TANIT93].

Cette étude nous a poussé à une modélisation des activités dans les USI qui nous permet de:

- définir les relations de dépendance entre les tâches requises et les rôles impliqués,
- bien comprendre les origines des «breakdowns» et leurs occurrences afin de les prévenir à temps,
- éviter les redondances dans le processus de travail.

Architecture globale

Les activités de soins en USI souffrent de plusieurs problèmes liés principalement à la mauvaise, voire l'absence, de coordination des tâches. Afin de pallier ces innombrables inconvénients, nous proposons une plate-forme coopérative distribuée.

L'approche des systèmes distribués se focalise sur la décentralisation des données et du contrôle [Elli91]. Essentiellement, ce type de système vise à maintenir la cohésion de l'état global en observant et en manipulant localement les paramètres. Pour respecter cette approche, notre architecture se divise essentiellement en un *espace partagé* contenant les objets d'information et un *espace distribué de contrôle*. Les objets d'information sont manipulés par:

- des acteurs (utilisateurs) en mode asynchrone au moyen d'un processus de conversation qui assure une communication structurée entre eux,
- des agents (processus autonomes) qui sont dotés de connaissances leur permettant de s'activer de façon opportuniste.

Le contrôle de l'architecture se décompose en deux types:

- le contrôle de cohérence, qui s'effectue sur les objets d'information, permet d'assurer leur intégrité et leur consistance via-à-vis de l'ensemble des informations partagées, et ce, en liaison avec des bases de connaissances et de données extérieures,
- le contrôle des interactions qui, basé autour d'un processus de conversation, permet de gérer la coordination et la communication entre acteurs. Il permet également de «tracer» toutes les actions accomplies sur l'information partagée.

Dans cette thèse nous n'allons pas traiter, de façon profonde, les problèmes liés à la distribution de l'architecture. Les recherches menées dans ce domaine sont nombreuses et intéressent beaucoup notre approche. Cependant, ces problèmes ne représentent pas le but

de notre travail de recherche. En revanche, nous nous focaliserons sur la modélisation et le contrôle des interactions au sein des organisations.

Modélisation

En considérant que les incidents se situent au niveau des interactions entre acteurs, nous nous sommes principalement appuyés, dans notre étude, sur:

- la théorie des actes de paroles de J. R. SEARLE [Sear69] [Sear87],
- la théorie de conversation de T. WINOGRAD et F. FLORES [Flor88] [Wino87a] [Wino87b],
- les travaux de T. MALONE sur la coordination [Malo88].

Ceci nous amène à privilégier les modèles qui sont basés sur la notion de transfert de responsabilités dans un processus de travail et qui, par conséquent, se focalisent sur les interactions humaines. Il s'agit des *modèles de Workflow*.

La théorie de workflow concerne l'étude et l'analyse des processus de travail, composés de tâches, événements ou interactions, qui créent ou ajoutent de la «valeur» dans les environnements de travail [Whit94].

Un workflow est donc un processus qui décrit le flux de travail. L'unité atomique d'un processus de travail est une boucle appelée *cycle workflow* [Wino87a] qui décrit les interactions entre deux acteurs ou agents au moyen des actions à réaliser, des rôles à jouer et des conditions à satisfaire.

Par ailleurs, le caractère générique du modèle de Workflow, tel que définit par WINOGRAD et FLORES [Wino87b], permet son application à toutes les organisations, à condition, bien-sûr, de définir les transactions entre participants et les rôles qu'ils seront amenés à remplir. Ce modèle «souffre», toutefois, de quelques problèmes liés à l'implantation physique du modèle. Par conséquent, bien que l'approche soit très intéressante pour des organisations où l'accent est particulièrement mis sur les utilisateurs, le modèle, en réalité, ne peut pas être applicable tel quel à des organisations comme les USI.

En effet, dans ce type d'organisation, le groupe dénote une entité ouverte, c'est-à-dire un ensemble de personnes dont les rôles priment sur les individus eux-mêmes. Le couplage entre acteurs et tâches est faible. Force est de constater que cette particularité rend ce modèle quelque peu rigide. Une extension du modèle s'avère donc nécessaire afin de l'adapter, en particulier, aux unités médicales et, en général, aux organisations dont les individus sont soumis à de fortes contraintes de mobilité aussi bien temporelle que spatiale.

Modèle PLACO

Nous avons défini un modèle, dérivé de celui de Winograd et Flores [Wino87b], appelé **PLACO**. Basé sur la notion de rôle, il permet de supporter *dynamiquement* le ou les acteur(s) requis pour accomplir une certaine tâche. Pour ce faire, nous avons identifié quatre types d'interactions fondamentales entre le personnel médical. Il s'agit de la *prescription* des soins, la *planification* des ressources humaines et matérielles, la *dispensation* ou l'administration des soins et l'*évaluation* clinique des soins. Ces quatre phases représentent un cycle fermé. Le passage d'une phase à une autre se fait via une *transition* pour assurer le bon déroulement des tâches.

On définit alors quatre transitions pour le cycle workflow qui sont: *l'acceptation des décisions* prises dans la première phase, *l'acceptation du plan d'action* et la spécification des conditions de satisfaction, *l'acceptation du rapport d'activités* médicales où les résultats obtenus ainsi que les ressources utilisées sont décrits et enfin, *l'acceptation de l'évaluation clinique*, i.e. la confrontation des résultats escomptés avec ceux réellement obtenus.

Afin d'éviter des situations de blocage en USI, nous avons étendu notre modèle pour qu'il puisse supporter les demandes de *délégation* et de *clarification* de la part des différents acteurs.

Pour mettre en évidence l'aspect fonctionnel du modèle dans les USI, nous avons introduit des messages semi-structurés dans le modèle abstrait. Ces messages forment un support de coordination entre acteurs puisqu'ils indiquent les conditions d'accomplissement d'une tâche donnée et l'état de l'information dans le cycle Workflow.

Les messages semi-structurés, comme l'a souligné T. MALONE dans [Malo86a], sont particulièrement utiles et efficaces pour les systèmes dédiés à supporter la coordination de tâches. Dans notre approche, les messages sont définis comme étant des «*post-it*» qu'on utilise communément pour annoter une information particulière et la transmettre à autrui afin d'améliorer la communication et la coordination entre individus.

Cette métaphore permet de comprendre aisément notre motivation à vouloir introduire la notion de coordination par un moyen des plus communs de nos jours qui est le «*post-it*». Ce faisant, nos «*post-it*» sont de plusieurs catégories: les «*post-it*» dits de *workflow*, les «*post-it*» de coordination et ceux dits libres.

Réalisation

Afin de valider notre modèle de façon concrète, nous avons réalisé un prototype qui n'est pas destiné à être installé dans les USI, mais qui permet de valider l'idée d'introduire un outil informatique pour supporter la coopération et la coordination dans des organisations telles que les USI. Pour des soucis de commodité de prototypage et de respect de l'approche orientée-objet de PLACO, nous avons opté pour une implémentation en *VisualSmalltalk*[®], version 3.0 sous *Windows for WorkGroups*[®]. L'avantage d'un tel langage apparaît dans sa grande aisance et sa rapidité à prototyper des systèmes orientés-objets. Ceci permet de valider les modèles et d'obtenir un canevas solide pour une réelle application.

L'application réalisée se compose de deux grandes parties séparément implantées:

- une partie représentant les objets d'information. Un objet d'information est un objet qui peut être vide, prescrit, planifié, dispensé ou évalué. Ces cinq étapes décrivent le cycle de vie d'un objet d'information. Chaque étape est réalisée par un acteur via une interface graphique,
- une autre partie qui représente le contrôle. Elle comporte trois agents: agent d'interface, agent de workflow et celui des «*post-it*».

La séparation du contrôle et du domaine présente plusieurs avantages. Elle permet:

- d'avoir une **application évolutive**. En effet, sans modifier les objets d'information, on peut ajouter aux agents des connaissances de contrôle supplémentaires sans modifier pour autant le fonctionnement des «*post-it*».

- de réutiliser le prototype réalisé pour un autre type d'application tels que l'agencement des réunions, des projets au sein d'une organisation par exemple, etc. Il suffit de modifier les règles de déclenchement des agents ainsi que le contenu des «post-it».

En résumé, l'introduction de nouvelles technologies pour la coordination des activités et la gestion des conflits dans les unités médicales requiert la prise en considération de plusieurs facteurs de nature distincte tant au point de vue humain que matériel ou organisationnel. La complexité d'un tel domaine nous a amené à décomposer le projet PLACO en quatre étapes:

1. *Une étude cognitive qui permet d'analyser le travail au sein des unités de soins intensifs, en particulier les relations d'interaction entre soignants dans un service de réanimation.*
2. *L'élaboration d'un modèle générique qui tient en compte la dimension humaine et décrit les interactions et échanges conversationnels entre acteurs de soins en intégrant les contraintes particulières aux USI. Le modèle que nous proposons est inspiré de la théorie des actes de langage de J. SEARLE [Sear69] et de la perspective du LangageAction de WINOGRAD & FLORES [Wino87a].*
3. *L'implantation du modèle en vue de la réalisation et l'évaluation d'un prototype.*
4. *La validation du modèle au regard des utilisateurs.*

Cette thèse s'intéresse donc à la modélisation des interactions humaines suivant les modèles orientés conversation dans des organisations à faible couplage entre individus. Elle nous permettra de cerner les problèmes liés à la coordination des tâches entre participants. Le domaine d'application choisi est la PLANification COopérative en unités de soins intensifs, d'où le nom du projet PLACO.

Afin de détailler chaque point, nous proposons, ci-dessous, le plan général de la thèse.

Plan général de la thèse

Cette thèse sera décomposée en deux grandes parties:

- I. **La première partie** concernera des généralités et se décomposera en trois chapitres traitant respectivement:

1. de la théorie du travail coopératif comme concept social, du CSCW comme discipline scientifique et des Groupware comme application de la technologie CSCW,
2. du Workflow (définitions et fondements), des concepts de la coordination et de la conversation, et des différentes applications et outils attenants,
3. de l'informatique médicale comme domaine de recherche relativement récent ainsi que des systèmes d'information dédiés à gérer l'information médicale.

II. La deuxième partie traitera de la problématique de la coopération et de la coordination dans les unités médicales, ainsi que de la solution que nous proposons en terme de modèle coopératif. Cette partie sera divisée en quatre chapitres:

1. Le premier chapitre élucidera le problème de coordination des tâches et activités médicales au sein d'une unité de soins intensifs. Pour ce faire, nous nous baserons sur l'étude et l'analyse psycho-sociologiques réalisées, conjointement avec le LABACOLIL, en 1992 dans une USI du CHRU de Lille. *Dans ce chapitre, nous proposons une architecture logicielle coopérative qui supporte les interactions des acteurs.*
2. Le deuxième chapitre sera consacré au modèle workflow asynchrone que nous proposons comme solution de modélisation du contrôle des interactions humaines, afin de remédier aux problèmes de coordination en USI. Dans un premier temps, nous présentons le modèle abstrait et ensuite, le modèle exécutif en incluant les messages comme mécanisme de coordination efficace.
3. Dans le troisième chapitre, nous introduirons la notion de «post-it», d'abord, comme solution aux problèmes de coordination et de communication au sein des USI, ensuite, comme métaphore de conception. Mais tout d'abord, nous parlerons du «tableau blanc» utilisé réellement dans les USI. Pour ce faire, nous essayerons de décortiquer les éléments qui composent ce tableau, le fonctionnement des «post-it» et leur contribution à la coordination des tâches. Ensuite, nous présenterons le «*post-it informatique*» ainsi que la représentation orientée objets de la solution proposée.
4. Le dernier chapitre sera consacré à la réalisation et à la validation du prototype de PLACO en VisualSmalltalk. Nous détaillerons le graphe d'héritage des objets et classes du prototype, ainsi que leur fonctionnement. Nous inclurons dans ce cha-

pitre un retour d'expérimentation de la part des utilisateurs potentiels, à savoir les médecins et les infirmières. Enfin, nous donnerons les perspectives liées à ce travail de recherche en terme de modélisation des interactions humaines.

PARTIE I

CSCW et Informatique Médicale

L'introduction de nouvelles technologies pour la coordination des activités et la gestion des conflits dans les unités de travail requiert la prise en charge de plusieurs facteurs distincts tant d'un point de vue humain que matériel ou organisationnel. La complexité inhérente au travail coopératif des groupes au sein des organisations nous conduit à décomposer cette partie en trois chapitres:

1. Le premier chapitre traitera des généralités liées au travail coopératif: théorie et fondements, et aux applications logicielles dédiées au groupes d'acteurs.
 2. Le deuxième chapitre décrira la théorie du workflow et les applications attenantes. Le paradigme de workflow appliqué aux unités de travail afin de résoudre les problèmes liés à la coordination.
 3. le troisième chapitre présentera le domaine de l'informatique médical sous un aspect orienté information. Nous décrirons les problèmes liés à la gestion de l'information dans les structures hospitalières, à l'intégration des outils informatiques dans un même système d'information et au problème de l'informatisation des unités de soins.
-

1.1 Introduction

Aujourd'hui, les autoroutes de l'information sont devenues réalité. Les architectures logicielles et matérielles appropriées ont également émergé et sont enfin disponibles. A partir de cette constatation, on peut en inférer que chacun d'entre nous sera bientôt capable de s'engager dans une forme particulière du *travail coopératif* supporté par ordinateur. Le Travail Coopératif Assisté par Ordinateur (TCAO) connaît, à l'heure actuelle, un vif essor dans le monde de la recherche. Qualifié parfois de science à part entière, il fait référence à un ensemble d'individus travaillant ensemble pour un même produit (un but commun) et utilisant comme support l'ordinateur. Ce domaine de recherche est aussi connu sous le nom de collaboration assistée ou supportée par ordinateur.

Plusieurs types de logiciels y sont consacrés, tels que les Groupware¹, les Workflow (Flux de Travail) et les Group Decision-Support Systems (les Systèmes Supportant la Décision en Groupe). C'est là un domaine de recherche et de développement vaste, fertile et prometteur. Cependant, beaucoup d'efforts restent à faire pour que le CSCW sorte du domaine de la recherche pour trouver sa place dans le quotidien. Il faut en effet:

- disposer de modèles de représentation des dynamiques sociales des activités de groupes,

1. Le terme «Groupware» est utilisé, ici, comme synonyme de «Collecticiel» tel que le préconise l'AFCEC. Il sera détaillé ultérieurement.

- développer des applications dont l'utilisation est intuitive,
- standardiser les termes usités au sein de la communauté «CSCW»,
- faciliter les interactions, souvent complexes, entre des tâches multiples réalisées par des groupes multiples.

Par ailleurs, les terminologies sont souvent différentes, voire contradictoires, selon la vision de chaque concepteur de systèmes CSCW [Grud94].

Dans la suite de ce chapitre, nous tenterons de développer et d'éclaircir les différents axes de recherche en matière de CSCW et de groupware. Nous éluciderons, par la même occasion, les différentes questions qui concernent le «comment» et le «pourquoi» de la grande famille CSCW. Mais d'abord, définissons le CSCW.

1.2 Le CSCW et le Travail Coopératif

1.2.1 Définition

Le CSCW ou le Travail Coopératif Assisté par Ordinateur¹ (TCAO) est un domaine de recherche multi-disciplinaire relativement nouveau. Il vise à étudier et définir des théories régissant le travail coopératif au sein des groupes et à concevoir des systèmes informatiques supportant les activités de groupes dans les organisations [Grud94]. En plus des sciences de gestion et de la sociologie, ce domaine de recherche rallie diverses approches de l'informatique: la communication, le traitement de données, les interfaces homme-machine, les systèmes d'information, etc.

1.2.2 Les premiers pas du CSCW

Le CSCW a émergé dans les années 80 grâce aux efforts développés de concert par les concepteurs industriels et les chercheurs dans différentes disciplines. Aujourd'hui, le grand effort reste à faire. Il s'agit de surmonter les difficultés inhérentes aux interactions multi-disciplinaires.

1. Le terme préconisé par l'AFCEC et qui dénote la même signification que le CSCW est les Activités Multi-Participantés.

La première approche du travail coopératif et des groupware s'est faite par l'intermédiaire de la bureautique. Les problèmes fondamentaux soulevés à l'époque n'étaient pas seulement d'ordre technique. Il s'agissait plutôt d'incompréhensions en matière de besoins des systèmes. Au milieu des années 70, les micro-ordinateurs promettaient de supporter le travail des groupes et des organisations de façon plus interactive et plus sophistiquée. Ainsi, la bureautique est née. Elle avait pour but primordial d'étendre et d'intégrer les applications mono-utilisateur telles que les traitements de texte et les tableurs, afin de supporter les groupes. *Mais quels sont les besoins exprimés pour ces systèmes?*

Faire avancer la technologie n'était pas suffisant. Les concepteurs de la bureautique avaient besoin d'apprendre davantage sur la façon dont les gens travaillaient au sein des groupes et des organisations, et la manière dont la technologie affectent leurs comportements. Quelques pionniers dans le domaine du CSCW, tel que D. ENGELBART [Enge63], ont longuement souligné ce point primordial. D'autres, spécialisés dans les systèmes de gestion d'information (Management Information Systems), prônent cette approche et la qualifient de moyen indispensable à la réussite des systèmes d'information.

La recherche dans le domaine CSCW a vu le jour grâce aux efforts déployés conjointement par les informaticiens, les économistes, les socio-psychologues, les anthropologistes, les théoriciens des organisations, les spécialistes de l'éducation, et tout autre chercheur spécialisé dans la théorie des activités de groupe. Ces efforts ont porté leurs fruits et se sont concrétisés dans plusieurs domaines d'application tels que la conception/fabrication assistée par ordinateur (CAD/CAM), l'ingénierie des logiciels, (CASE), l'ingénierie concurrente (CE), la gestion du Workflow (WFM), le télé-enseignement, la télé-médecine, les conférences temps-réel, etc.

1.2.3 Le groupe et le CSCW

Le CSCW est vraisemblablement une des rares disciplines où les scientifiques s'efforcent de «construire» une passerelle entre plusieurs domaines de recherche autour d'une seule notion qu'est le *groupe*. Les questions fondamentales auxquelles ils tâchent de répondre peuvent se récapituler comme suit:

- Quels sont les besoins à satisfaire en termes de modèle et de technologie pour la conception des systèmes supportant le groupe?

- Quelle est la tâche du groupe?
- Quels sont les moyens et les modes de communication à utiliser?
- Comment le processus de groupe est-il distribué parmi les sous-groupes?

Toutes ces questions exigent une compréhension parfaite de la notion de groupe de travail et du travail coopératif. Une modélisation de groupe s'avère donc plus que nécessaire.

Dans le domaine de la santé, la notion de groupe fixe et fermé n'a plus beaucoup de sens. On raisonne plutôt en terme d'*équipe ouverte*. En effet, les acteurs de soins, en particulier les médecins, sont très mobiles dans l'espace et dans le temps et peuvent faire partie de plusieurs équipes indépendantes. Ils peuvent donc être remplacés, chaque fois que le besoin se fait sentir, par d'autres personnes de compétence équivalente.

Cette contrainte de mobilité intrinsèque, qui est plus ou moins forte, augmente la complexité de la coopération et par conséquent, celle du travail coopératif. Dans ce domaine précis, l'accent doit être mis sur le rôle joué par l'individu au détriment de la personne elle-même. Cette réflexion, souvent omise par les concepteurs de groupware, est déterminante et montre combien la notion de groupe est fondamentalement complexe et que la conception d'un outil supportant le travail de ce dernier l'est encore plus.

1.2.4 Les concepts-clés du CSCW

Depuis plusieurs années, l'informatique, et particulièrement le domaine d'étude des relations homme-machine, s'est axé autour de l'interaction entre un individu et un système informatique. Dans cette discipline, qui concernent les interfaces homme-machine, beaucoup d'efforts ont été déployés favorisant ainsi l'émergence de deux grandes approches:

1. *Le paradigme linguistique* où l'ordinateur est considéré comme un correspondant artificiel.
2. *Le paradigme «objet»* où des actions sur un micro-monde sont possibles. L'ordinateur est considéré alors comme un «théâtre».

Mais l'émergence du CSCW a imposé un changement de paradigme. Le travail est considéré comme un phénomène social [Schm91]. Ce qui est devenu important, c'est la communication homme-homme médiée par les télécommunications et les systèmes d'information.

1.2.4.1 Le travail d'articulation

Dans une coopération, chaque effort induit des tâches secondaires de médiation et de contrôle de l'association des individus. Ceci nécessite un *travail d'articulation* qui permet de coordonner les activités de l'ensemble des individus.

La majorité des systèmes traditionnels sont conçus en se basant sur l'idée que le travail concerne simplement le flux d'information. Ce concept simpliste néglige le travail d'articulation qui est indispensable quant au bon déroulement du processus de travail. En revanche, les systèmes CSCW supportent l'auto-organisation des individus. Ils leur permettent de négocier librement l'allocation et l'articulation des tâches.

1.2.4.2 Les espaces d'information partagés

Le partage d'information est le problème central des systèmes CSCW. Comment partager un espace d'information entre différents utilisateurs, dans différents endroits, avec différents rôles et appartenant à différents groupes?

L'objectif visé est de partager la vision du monde de l'information des utilisateurs avec autrui, à travers les manipulations et les transformations de leur propre situation. Ceci implique trois concepts différents:

1. le concept de *l'environnement multimedia partagé* (ensemble d'objets son ou image),
2. le concept de *point de vue*. Chaque individu a un point de vue différent d'une même information,
3. le concept du *matériel partagé*. La coordination du travail coopératif est souvent nécessaire. L'ordre des tâches est essentiel.

1.2.4.3 Les espaces de conversation

Les systèmes CSCW apportent aux concepteurs de nouveaux problèmes liés à l'intégration des bases de données et des systèmes de communication. Ces problèmes incluent l'accès concurrent à l'information, la communication inter-personnelle, etc. Plusieurs travaux ont

déjà montré l'importance des théories de conversation pour une meilleure compréhension de la coopération entre individus. Ces travaux incluent la théorie des actes de langage de AUSTIN et SEARLE [Aust62] [Sear69], et celle du Workflow de WINOGRAD et FLORES [Wino87b].

1.2.5 Les formes de coopération

«Pourquoi les gens coopèrent?» ou *«Pourquoi des processus de production sont-ils effectués de manière coopérative?»*.

Les individus coopèrent à cause des capacités limitées de chaque individu [Schm91]. On considère essentiellement trois types de coopération:

1.2.5.1 La coopération additive

Les capacités humaines individuelles sont limitées, sur un plan physique ou physiologique, mais aussi au plan psychologique. SIMON [Simo57] avait mis en évidence cette «rationalité limitée» des processeurs d'information humains, ce qui avait amené INBAR [Inba79] à cette conclusion désenchantée: *«Intellectuellement, l'homme apparaît comme un ordinateur particulier, caractérisé par une capacité de traitement de l'information lente, limitée et séquentielle»*.

Cette capacité limitée peut être compensée par une «coopération additive»: en combinant capacités et efforts, un ensemble d'individus peut effectuer une tâche qui aurait été impossible à un seul individu. Paradoxalement, les innovations technologiques, en augmentant les capacités physiques et les possibilités de traitement de l'information de chacun, ont réduit l'impact de la coopération additive. NEWTON [Newt85] conclut que *«les innovations technologiques ont rendu les activités humaines plus efficaces et chacun plus puissant, et les activités coopératives (au sens de la coopération additive) ne sont plus aussi efficaces»*.

1.2.5.2 La coopération intégrante

Avec la spécialisation croissante de chacun, il apparaît évident que chaque individu ne peut manier avec succès qu'un nombre limité d'outils. Il peut s'agir de deux types d'outils:

- les outils physiques ou matériels,

- les outils intellectuels qui sont plus difficiles à cerner. Ce sont des compétences, des connaissances ou des expertises.

Cependant, dans l'un ou l'autre cas, l'évolution actuelle s'effectue vers une spécialisation croissante et chacun se limite à un domaine d'activités. Cette différenciation du travail nécessite une *activité coopérative d'intégration*.

L'introduction d'un système de contrôle et d'intégration des activités augmente la capacité de chaque travailleur individuel, en transférant le mécanisme de contrôle à la machine, tandis que les travailleurs assurent la planification et la supervision des tâches, ainsi que le diagnostic de pannes. Mais il faut que les travailleurs aient une bonne connaissance du processus de contrôle qui gère l'organisation du travail, afin d'en détecter les défaillances et d'en optimiser l'efficacité [Schm91].

1.2.5.3 La coopération de débat

La validité et la véracité d'une décision obtenue par un processus de travail basé sur la connaissance est fragile et contestable [Inba79]. L'objet de coopération est de répondre à cette insuffisance.

Le fait que le travail soit basé sur la connaissance entraîne l'existence de plusieurs stratégies possibles. L'existence de ces différentes stratégies s'explique par des «*points de vue*» divers quant à la solution à apporter à un problème donné.

La coopération de débat sert à sélectionner les bons ajustements, les bonnes solutions parmi l'infinie variation des points de vue. Même s'il existe des décisions erronées et des erreurs d'interprétation, des individus qui débattent de leurs problèmes avec leurs collègues de travail arrivent généralement à des décisions objectives et équilibrées dans des environnements complexes.

1.2.6 Les espaces de la coopération

Le travail coopératif s'organise à travers l'espace et le temps. On peut donc, selon la classification de JOHANSEN [Joha88], considérer les processus de collaboration selon deux axes principaux: la coopération locale ou distante, synchrone ou asynchrone.

1.2.6.1 Coopération locale ou à distance

Les travailleurs qui coopèrent dans un même lieu (dans la même pièce, par exemple) sont en mesure d'interagir fréquemment, tandis que ceux qui coopèrent à distance sont limités dans leurs interactions par la disponibilité et la largeur de la bande passante, ainsi que par le temps de réponse du moyen de communication utilisé.

RODDEN [Rodd91] énumère quatre catégories de systèmes destinées à supporter le travail d'un groupe selon leur nature géographique (c.f. figure 1):

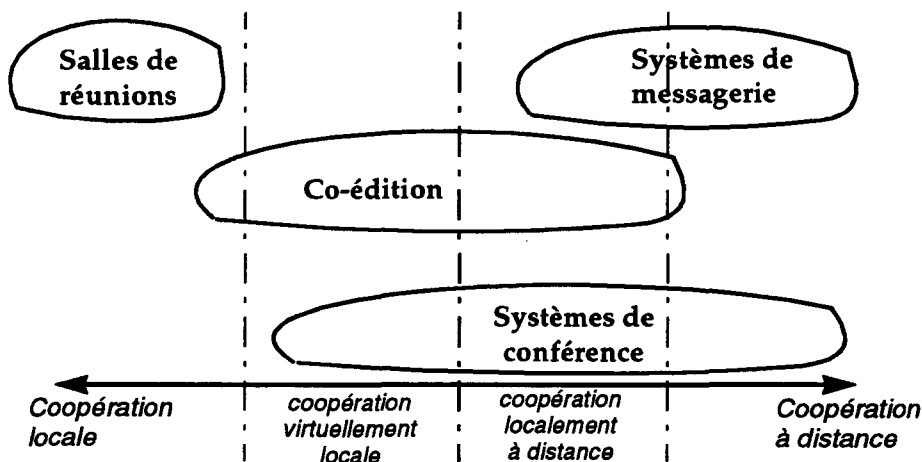


FIGURE 1 La nature géographique dans les systèmes CSCW.

1. *systèmes supportant la coopération locale.* Ils nécessitent la présence des utilisateurs dans un même endroit.
2. *Systèmes supportant la coopération virtuellement locale.* Les utilisateurs ne sont pas présents dans le même endroit. Ils le sont virtuellement grâce à des liens temps-réel de type audiovisuel. Cette classe inclut les systèmes de conférence multimedia en temps-réel tels que **MMConf** [Crow90] et **MERMAID** [Wata90].
3. *Systèmes supportant la coopération localement à distance.* Ils offrent un accès en temps-réel à très large bande entre utilisateurs. Ces systèmes utilisent souvent les techniques de partage d'écrans. Cette catégorie comprend les systèmes de co-édition tels que **CoAuthor** [Hahn91] et **gIBIS** [Conk87].
4. *Systèmes à distance.* Ils offrent un accès minimum entre les utilisateurs. Cette classe inclut les systèmes de messagerie classique.

1.2.6.2 Coopération synchrone ou asynchrone

Les différentes tâches et sous-tâches d'un travail coopératif peuvent être effectuées simultanément ou reportées dans le temps. L'intervalle de temps entre deux tâches coopératives varie d'autant. Les sous-tâches peuvent être réalisées comme une suite d'actions étroitement couplées ou comme une série d'actions interconnectées.

Comme illustré par la figure 2, il existe quatre principales classes des systèmes CSCW qui sont destinées à supporter la coopération synchrone et celle asynchrone [Rodd91].

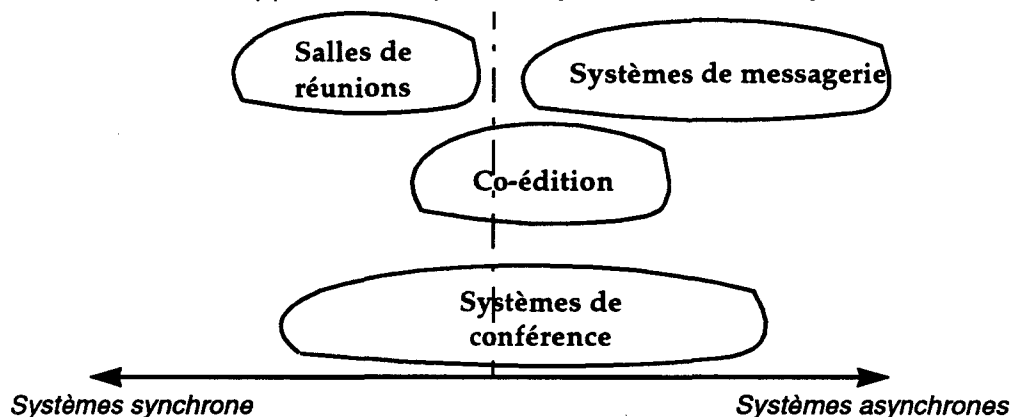


FIGURE 2 Classes des systèmes CSCW supportant la coopération synchrone et asynchrone.

1.2.6.3 Coopération collective ou distribuée

Dans le mode collectif du travail coopératif, les individus coopèrent ouvertement et consciemment: ils constituent un groupe qui a une responsabilité commune.

En revanche, dans le mode distribué, les individus sont semi-autonomes. Chacun peut modifier son comportement selon les circonstances et avoir sa propre stratégie: dans cette situation, chaque travailleur n'est pas nécessairement conscient des autres ni de leurs activités; ils coopèrent au travers de leur espace de travail.

1.2.6.4 Coopération directe ou indirecte

Dans le cadre de la coopération directe, les travailleurs interagissent en échangeant une information symbolique: ils communiquent.

Dans la coopération indirecte, les individus coopèrent via une machine. SCHMIDT [Schm91] illustre ce mode de coopération par un exemple d'un travailleur A qui, par

rapport à l'état de fonctionnement d'une machine, va prendre une décision. Celle-ci va modifier l'état de la machine. En fonction de ce nouvel état, un travailleur B peut prendre une autre décision qui va entraîner une nouvelle modification. Dans ce cas, les travailleurs ne communiquent pas. Néanmoins, ils coopèrent.

1.2.7 Les systèmes répartis et le CSCW

A cause des utilisateurs qui sont «distribués» dans le temps et/ou dans l'espace, plusieurs systèmes multi-utilisateurs sont considérés naturellement distribués [Elli91]. L'approche de tels systèmes se focalise sur la décentralisation des données et du contrôle. Essentiellement, ce type de système vise à maintenir la consistance de l'état global en observant et en manipulant localement les paramètres.

L'analyse des algorithmes efficaces pour les systèmes d'exploitation distribués et les bases de données distribuées représente un vaste domaine de recherche dans la théorie des systèmes distribués. Par exemple, implémenter un système de messagerie électronique induit des problèmes de distribution complexes liés à la robustesse et la fiabilité de ces systèmes.

Les systèmes distribués (SD) ont connu une émergence rapide durant les dix dernières années. Les produits tel que ETHERNET [Shoc82] se sont stabilisés et répondent au standard international. Les systèmes distribués entrent actuellement en phase de stabilité et de consolidation.

Cependant, le problème majeur lié à la coopération dans les systèmes distribués est le besoin incessant de reconnaître l'autonomie des sites de travail sur le réseau. En effet, une forte coopération et une grande autonomie se situent aux deux extrêmes du spectre dans lequel sont classés les systèmes distribués [Rodd91]. En augmentant l'autonomie d'un système, on diminue inévitablement le degré de coopération et inversement [Rodd91].

Traditionnellement, le contrôle dans les systèmes CSCW distribués se base sur l'approche orientée système. Dans ce cas, le contrôle traite les problèmes liés à la distribution de l'application et à la transparence de cette distribution. Malheureusement, le fait de se focaliser sur la transparence a forcé les concepteurs à utiliser l'approche *bottom-up* pour le développement des systèmes distribués (approche ascendante). Le partage est

transparent pour chaque utilisateur. Ainsi, chaque individu est inconscient des activités des autres. Ce qui contredit totalement les principes du CSCW.

Le terme de «*distribution transparente*» dénote plusieurs sens. Nous proposons, dans le tableau 1, les différentes formes de transparence [Rodd91].

Transparence	Problème Central	Résultat de la Transparence
Endroit	L'emplacement d'un objet dans un environnement distribué.	L'utilisateur est inconscient de l'endroit où se trouvent les services
Accès	La méthode d'accès aux objets dans un système distribué.	Tous les objets sont accessibles de la même façon
Migration	L'emplacement d'un objet dans un environnement distribué.	Les objets peuvent se déplacer sans que l'utilisateur en prenne conscience.
Concurrence	L'accès partagé aux objets dans un environnement distribué.	Les utilisateurs ne traitent pas les problèmes des accès concurrents aux objets.
Duplication	Le maintien des copies d'un objet dans un environnement distribué.	Le système traite la consistance des copies des données.
Panne	Les pannes partielles dans un environnement distribué.	Les problèmes des pannes sont masqués pour l'utilisateur.

TABEAU 1 Les formes de transparences dans les systèmes distribués.

Les systèmes d'exploitation distribués (Distributed Operating Systems) tentent de mettre en place chacune de ces formes de transparence. Cette approche n'est pas adaptée aux systèmes CSCW distribués, car le résultat revient à figer le contrôle dans le coeur du système. Le contrôle ne peut donc être modifié ou paramétré selon les classes d'application. A cause de l'aspect dynamique des systèmes CSCW, il est improbable que cette approche soit adaptée.

Afin de résoudre ces problèmes, plusieurs approches intermédiaires ont été explorées:

1. *La non-transparence.* Dans les systèmes non-transparents, la distribution est visible pour l'utilisateur. Ils doivent donc supporter eux-même les problèmes telles que les pannes et la migration. Ce type de système offre plus de flexibilité aux utilisateurs puisqu'ils peu-

vent manipuler les objets dans un environnement distribué. En revanche, la distribution peut s'avérer très lourde pour le programmeur.

2. *La transparence sélective.* Le concepteur de l'application possède le choix de sélectionner, parmi les formes de transparence énumérées dans le tableau 1, celles qui sont adaptées à l'application. Cette approche offre plus de flexibilité mais n'est pas vraiment adaptée aux systèmes CSCW, car l'utilisateur n'a pas la possibilité de modifier dynamiquement la transparence du système.
3. *L'adaptabilité des règles.* Les règles de distribution offrent un contrôle explicite dans les systèmes CSCW, et un degré de liberté supplémentaire aux utilisateurs. Comme le décrit ARMSTRONG [Arms90], «*Les règles sont à la fois restrictives et permissives. Elles tra-cent les limites d'action, mais, en même temps, offrent la liberté d'agir à l'intérieur des limites*».

1.3 Le Groupware

Dans cette partie, nous examinerons une question pertinente: Que doit-on inclure dans la rubrique des applications CSCW?

Bien que l'ordinateur soit à présent banal sur les lieux de travail, sa présence n'a guère modifié les relations et les contacts entre individus. Mais les technologies de l'informatique et de la communication continuent à converger et cette évolution favorise, d'une part, les collaborations entre individus distants, et d'autre part, la coordination des tâches qu'ils exécutent au sein d'un véritable «*espace de travail électronique*».

Un aboutissement de ce mariage technologique sera peut-être le lieu de travail électronique (i.e. un système à large organisation intégrant le traitement de l'information et les activités de communication). L'étude de tels systèmes relève du CSCW. Les produits commerciaux du CSCW, tels que **COORDINATOR** [ATI89] et autres logiciels PC, sont souvent cités comme des exemples de groupware.

1.3.1 Définition du groupware

Ce terme est souvent utilisé comme synonyme de la technologie CSCW. D'autres définissent le groupware comme un logiciel dédié aux petits groupes de travail et non aux grandes organisations. D'après C.A. ELLIS [Elli91], «*le groupware peut être vu comme une*

classe d'application pour les petits groupes et les organisations, combinant des ordinateurs et des bases d'information avec la technologie de la communication. Ces applications ne supportent pas forcément la coopération».

La plupart des systèmes logiciels supportent uniquement l'interaction entre un utilisateur et le système, que ce soit pour préparer un document, exécuter des requêtes sur une base de données ou même manipuler un jeu vidéo. Même les systèmes supportant les applications multi-utilisateurs, tels que les systèmes de gestion d'information, gèrent à peine les interactions de type utilisateur-utilisateur. Nous donnerons, ci-dessous, la définition d'un groupware, qu'il faudra retenir pour la suite et qui est inspirée de C.A. ELLIS [Eli91]:

«Le but d'un groupware est d'assister les groupes dans la communication, la collaboration et la coordination de leurs tâches. Les groupware sont donc des systèmes informatiques à base d'ordinateur, qui supportent des groupes d'utilisateurs engagés dans une tâche (ou but) commune et qui fournissent une interface pour un environnement partagé».

Retenons deux notions capitales de cette définition. Il s'agit des termes «*tâche commune*» et «*environnement partagé*». Ces derniers excluent les systèmes multi-utilisateurs tels que ceux à partage de temps où les utilisateurs ne partagent pas de tâche commune. Remarquons aussi que la définition ne spécifie pas la concomitance des utilisateurs dans leur travail. Les groupware supportant des activités simultanées sont appelés *groupware en temps-réel*.

Le groupware, tel qu'il est défini par JOHNSON-LENTZ [John82], signifie un système informatique basé sur l'ordinateur et englobant des processus de groupes sociaux. Par contre, JOHANSEN [Joha88] se restreint, dans sa définition des groupware, aux systèmes informatiques dont le support est l'ordinateur. Dans la pratique, le succès de tels systèmes dépend de l'équilibre délicat entre les bons processus sociaux et la technologie appropriée.

1.3.2 Typologies des systèmes groupware

Les systèmes groupware sont de plus en plus nombreux. ELLIS propose une classification de ces derniers suivant leur niveau d'application [Eli91]. JOHANSEN suggère une autre taxonomie des groupware en considérant les dimensions de l'espace et du temps [Joha88].

1.3.2.1 Classification selon le niveau d'application

1. **Les systèmes de messagerie** qui supportent les échanges asynchrones des messages textuels entre groupes d'utilisateurs. Les nombreux exemples incluent, d'une part les systèmes de messagerie classique tel que **Email**, et d'autre part les systèmes de messagerie intelligente tels que «**Information LENS**» [Mal86b], qui permet aux utilisateurs de préciser les règles de filtrage et d'acheminement des messages suivant leur contenu, et le système «**Imail**» [Hogg85] qui permet, grâce à un langage de scripts, d'associer des programmes aux messages qui s'exécutent dans l'environnement du destinataire.
2. **Les éditeurs multi-utilisateurs** qui permettent à un groupe d'utilisateurs de composer et d'éditer des documents en commun. Certains systèmes, tels que **ForComment**[®] [Oppe88], sont destinés à une utilisation synchrone. Les systèmes de co-édition temps-réel permettent à un groupe d'utilisateurs d'éditer le même objet en même temps. Ils permettent donc un accès en lecture concurrent aux objets mais limitent, à un seul, l'accès en écriture. Les exemples nombreux de tels systèmes comptent «**Collaborative Editing Systems**» (CES) [Grei86], «**Shared Book**» [Lewi88], «**Quilt**» [Fish88] [Lela88], «**Mercury**» [Kais87], etc.
3. **Les systèmes de co-décision et salles de réunion électroniques** («*Group Decision Support Systems*», et «*Electronic Meeting rooms*»), proposent des moyens d'exploration des problèmes non structurés. Le but est d'améliorer la productivité des réunions de décisions de groupe. La plupart des **GDSS** sont implémentés sous forme de *salle de réunion électronique* contenant plusieurs machines interconnectées via un réseau et des équipements audiovisuels. L'exemple le plus connu est **PlexCenter Planning and Decision Support Laboratory** développé à l'université d'Arizona [Appl86]. La salle de conférence contient une table de conférence en forme de U avec huit stations de travail, un disque vidéo, et un grand écran de projection qui sert à visualiser les écrans des stations de travail des participants.
4. **Les systèmes de conférence:** Dans cette catégorie, l'ordinateur sert de médium de communication sous différentes formes. Ces systèmes se divisent en trois groupes:
 - «*Real-time Computer Conferencing*» qui permet à un groupe d'utilisateurs présent, soit dans une salle de réunion électronique ou dans des endroits différents, d'interagir en mode synchrone à travers leur station de travail.

- «*Computer Teleconferencing*» qui fournit un support de télécommunication (audio et vidéo) pour les interactions de groupe. Cette approche présente l'inconvénient de nécessiter des salles spéciales et, parfois, des utilisateurs entraînés [Joha84].
 - «*Desktop Conferencing*» qui est une approche intermédiaire entre les deux précédentes. Un exemple de système utilisant cette approche est **MMConf** [Crow90] qui fournit aux utilisateurs un écran partagé de documents multimedia et des canaux de communication pour la voix et les pointeurs partagés.
- 5. Les systèmes de coordination:** Le problème de coordination concerne l'intégration et l'ajustement harmonieux des efforts individuels vers l'accomplissement d'un but global [Elli91]. Les systèmes de coordination abordent ce problème de différentes façons. Typiquement, ils permettent aux utilisateurs de «voir» leurs actions ainsi que celles des autres s'accomplir dans un contexte partagé. Les systèmes de coordination se caractérisent par les modèles sur lesquels ils sont basés. C.A. ELLIS en dénombre quelques types:
- *modèles orientés formulaire* qui se basent sur le routage des documents et formulaires au sein d'une organisation. Les systèmes basés sur ces modèles abordent le problème de coordination en modélisant explicitement les activités par des processus fixes et rigides [Loch88].
 - *modèles orientés procédures* qui considèrent les procédures organisationnelles comme des processus programmables [Blaz89]. L'élaboration de ces processus se compose de plusieurs phases: spécification, conception, implémentation et vérification.
 - *modèles orientés conversation* qui sont basés sur le fait que les individus coordonnent leurs activités via la conversation [DeCin86] [Flor88] [Wino87a]. Ces modèles reposent sur la théorie des actes de langage de J. R. SEARLE [Sear69] [Sear86]. L'exemple le plus connu est le système **COORDINATOR** [Flor88] basé sur un ensemble d'actes de langage tels que les demandes, les promesses, etc.
 - *modèles orientés communication structurée* qui décrivent les activités de l'organisation sous forme de relations entre rôles. Dans l'approche ITT [Holt88], l'environnement de travail électronique d'une personne est composé d'un ensemble de centres où chaque centre représente une fonction dont une personne est responsable. Au sein de chaque centre, il y a des rôles qui exécutent le travail et des objets qui forment le matériel nécessaire à la réalisation de la fonction du centre.

1.3.2.2 Classification selon l'espace et le temps

Les systèmes groupware sont conçus pour aider, dans ses activités, un groupe dont les membres sont face-à-face, ou un groupe distribué dont les individus sont situés dans des endroits différents. En outre, un système groupware peut être conçu pour améliorer la communication et la collaboration en temps-réel ou en temps différé. Ces considérations, à la fois spatiales et temporelles, divisent les systèmes groupware en quatre catégories que nous schématisons dans la figure 3.

Ainsi, les systèmes de salles de réunion électroniques se situent dans la cellule en haut à gauche; les systèmes de co-édition en temps-réel dans la cellule en bas à gauche; les systèmes de tableaux électroniques dans la cellule en haut à droite; enfin, les systèmes de messagerie dans la cellule en bas à droite.

Comme le suggère JOHANSEN dans [Joha88], on peut affiner la classification des groupware en incluant une autre dimension, telle que la taille du groupe par exemple.

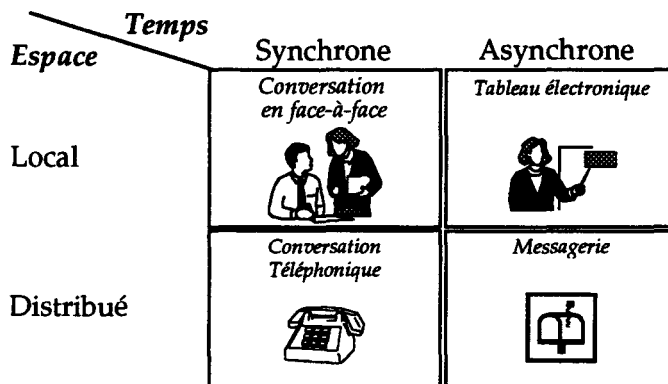


FIGURE 3 Taxonomie des activités de groupe suivant les dimensions de l'espace et du temps.

1.4 Conclusion

Certains auteurs considèrent le CSCW comme discipline scientifique. Cependant, d'après les expériences actuelles, le CSCW ressemble plus à un forum d'idées, d'observations, de problèmes et de technologies qu'à une discipline à part entière.

Le CSCW est un thème de recherche multi-disciplinaire impliquant l'ergonomie, l'anthropologie, la sociologie des organisations, la sociologie scientifique, l'étude des processus industriels, la théorie de coordination, la théorie de la décision, la psychologie cognitive, l'intelligence artificielle, les interactions homme-machine, etc. Cependant, la notion de travail coopératif est un concept très spécifique qui ne s'applique pas à tout système de production: il faut qu'il y ait production d'un produit particulier ou d'un ensemble de produits [Schm91].

Mais les systèmes de coopération ne sont pas souvent couronnés de succès [Grud88]. Parmi les nombreux facteurs responsables de l'échec des applications CSCW, deux ont retenu notre attention:

1. Le bénéfice n'est pas partagé. La disparité entre ceux qui bénéficient de l'application et ceux qui doivent faire le travail nécessaire pour assurer ce bénéfice. Il faut s'assurer que tout individu dans un groupe tire directement bénéfice lors de l'utilisation de l'application. Ceci implique qu'il faut éliminer ou minimiser le travail supplémentaire que certaines personnes sont amenées à fournir pour les autres.
2. L'extrême difficulté de l'évaluation de ces applications. L'évaluation des applications CSCW exige une approche différente de celle des applications mono-utilisateur. En effet, plusieurs facteurs, tels que la motivation, l'économie, la politique, entrent en jeu [Malo85]. L'évaluation de ces applications doit se baser sur des méthodologies nouvelles incluant la psychologie sociale et l'anthropologie.

En outre, l'évaluation des systèmes groupware dans leur champ d'application est relativement complexe. Cette difficulté est due principalement:

- au nombre de personnes à observer dans chaque site,
- à la grande hétérogénéité des groupes,
- aux facteurs environnementaux qui influencent l'acceptation tels que la formation des utilisateurs au système groupware, la lourde gestion qui en découle, etc.

Les groupware ont émergé du mariage de la technologie des communications avec l'ordinateur. Les problèmes liés à la conception de ce type de système sont nombreux et variés. Le partage d'information dans les groupware, par exemple, induit des problèmes non encore cernés, et qui sont liés aux aspects de distribution et d'interfaces-utilisateur. Bien que l'avenir des groupware semble être prometteur, nous devons tenir compte des échecs passés et expériences amères dont ont été victimes plusieurs systèmes supportant le travail de groupe, tels que les systèmes de gestion des calendriers [Grud88].

L'utilisation de l'ordinateur comme support des activités des individus dans leur groupe et leurs contextes organisationnels, changera indiscutablement et profondément la manière de travailler, voire de vivre, des gens [Grud88]. Nous avons donc besoin de bien comprendre et cerner les modes de fonctionnements des groupes et des organisations pour faire évoluer les systèmes CSCW de manière efficace.

Au même temps, nous nous devons d'appréhender les différences de culture des individus et leurs réactions face à une nouvelle technologie si l'on veut concevoir des systèmes efficaces dédiés à supporter des groupes entiers. Nous devons également développer un cadre conceptuel complet pour mieux comprendre nos propres comportements. Les intuitions qui nous ont guidés dans le passé sont révolues. Si l'on souhaite concevoir des systèmes supportant les groupes dans leur intégralité et leur diversité, nous devons apprendre plus sur la façon dont les différents types de personnes travaillent. Nous devons donc éviter au maximum de nous inspirer de nos intuitions car les expériences dans le domaine du CSCW sont coûteuses en termes d'argent et de temps.

Nous finirons ce chapitre par quelques questions fondamentales non encore élucidées:

1. Quels sont les mécanismes les plus aptes à supporter le travail de groupe?
2. Quelles sont les règles de contrôle adéquates pour le CSCW?
3. Comment représenter la coopération et le contrôle dans les systèmes CSCW?

Les réponses à ces questions exigent une bonne compréhension des comportements, à la fois, des systèmes distribués et des groupes d'utilisateurs.

2.1 Introduction

Alors que l'organisation du travail évolue, d'un modèle tayloriste qui renforce les structures hiérarchiques, à un modèle plus respectueux d'une nécessaire coopération de l'ensemble des travailleurs, les outils informatiques s'adaptent à ces nouveaux environnements. Le **Workflow** en est un exemple.

Aujourd'hui, il existe deux grandes définitions du **workflow** [Whit94]:

1. L'étude et l'analyse des processus de travail, composés de tâches, d'événements ou d'interactions, qui créent et augmentent «*la valeur*» dans les environnements de travail.
2. La technologie et les logiciels qui analysent ou gèrent les éléments constitutifs des processus de travail.

2.2 La Gestion de Workflow

Plusieurs organisations ont recensé les problèmes liés à la coordination du travail. Des procédures écrites sur papier ne sont pas ou partiellement utilisées en pratique; le travail est suspendu; les responsabilités sont mal définies et le personnel passe la majorité de son temps à corriger les actions et les procédures. Afin de pallier ces inconvénients, une meilleure compréhension des processus de travail est nécessaire. Jusqu'à présent, la coordination du processus est souvent réalisée en se basant sur l'intuition personnelle,

mais la modélisation et la mise à disposition de stratégies peut aider à la mise en place de véritables contrôles des processus de travail.

La gestion de workflow (abrégé WFM comme *WorkFlow Management*) est un terme récent utilisé principalement dans les domaines de la bureautique (OA), de l'administration des affaires (BM), de la communication de données (DC), des systèmes d'information (IS) et du CSCW.

Aujourd'hui devenue discipline de recherche, la WFM vise à améliorer la coordination du travail. La WFM est souvent utilisée dans les situations où les procédures administratives sont exécutées par plusieurs personnes distribuées dans l'espace (c'est-à-dire dans des endroits différents). L'amélioration de la coordination est possible par la disponibilité d'une infrastructure de communication électronique dans le lieu de travail [Elli91].

Les bénéfices attendus du WFM se résument en quatre points:

- la compréhension quantitative et structurale du processus de travail,
- la flexibilité dans le travail par la possibilité d'effectuer un changement rapide dans la structure et les paramètres des processus du travail,
- la simplicité des procédures pour le personnel,
- l'économie du temps.

2.3 La perspective du Langage/Action

T. WINOGRAD propose, suite aux travaux d'AUSTIN et de SEARLE sur la conversation, une nouvelle manière de voir les systèmes informatiques: *la perspective du langage/action*. Cette dernière représente la brique de base d'un des produits informatiques les plus connus de la famille CSCW qui est *COORDINATOR*[®], développé par *ATI: Action Technologies Incorporation*. Cette perspective montre la nécessité d'inclure, dans tout système informatique, le «jargon» ou *langage spécialisé* lié à chaque organisation. Un système qui ne supporte pas le jargon d'une organisation a de faibles chances d'être utilisé à large échelle.

Si on considère la perspective du *langage/action*, on dira que *les gens agissent à travers le langage (ou la parole)* [Wino87a]. Cependant, si on considère la perspective du traitement de l'information, on dira que *les gens traitent l'information et prennent des décisions*. Naturellement, n'importe quelle personne, dans une organisation, peut faire les deux. Toutefois, il existe une différence de fond entre les deux perspectives.

Un des résultats les plus intéressants qui ont émergé des travaux de T. WINOGRAD, concerne l'analyse des erreurs et des «écarts» qu'induit la perspective du *langage/action* [Wino87a].

Une façon utile d'identifier cette perspective est de se poser la question suivante: «*Que font les gens?*».

Considérons la situation de l'infirmière qui appelle la pharmacie, se renseigne sur la disponibilité des médicaments, et en commande un pour le patient. Dans la perspective du traitement de l'information, on peut se limiter à décrire la base de données contenant les informations sur les médicaments et sur les règles de décision de commande des médicaments. Dans l'autre perspective, celle concernant le *langage/action*, on se focalisera plutôt sur l'action de commander les produits et sur les modes d'interaction, telles que la conversation préliminaire concernant la disponibilité des médicaments et les conversations sous-jacentes qui se tissent au fil du processus lié à l'exécution de la commande. Dans d'autres perspectives, on pourrait considérer la relation personnelle entre le pharmacien et l'infirmière, la rentabilité (coût-efficacité) de la communication par téléphone ou encore, le statut légal des commandes effectuées par une infirmière.

Chaque perspective doit se référer à un schéma conceptuel et à une méthodologie. Pour des considérations de rentabilité (coût-efficacité), on doit se tourner vers la théorie de l'économie. Pour le traitement de l'information, on doit s'inspirer des théories de l'information et de prise de décisions. Pour la perspective du *langage/action*, on doit s'intéresser non pas au détail des paroles du langage naturel, mais plutôt aux problèmes de forme, de signification, d'interprétation et d'utilisation qui sont communs à la communication humaine, c'est-à-dire à la composante pragmatique du langage. Un exemple classique est la déclaration suivante: «Il fait froid ici!» prononcée par un *maître* à l'attention de sa *servante*. Bien que le sens littéral de la phrase indique une information sur

la température, il n'en reste pas moins évident que l'intention est de faire accomplir une action par la servante [Wino87a].

Notre principale attention, dans l'analyse du workflow, se focalise sur la *pragmatique* du langage comme un aspect fondamental du langage: son rôle dans le déclenchement et l'interprétation des actions.

2.3.1 Analyse conversationnelle

WINOGRAD et FLORES formalisent le cours élémentaire d'une conversation pour action par un diagramme d'état-transition (c.f. figure 4). Chaque rond représente un état possible de la conversation. Quant aux flèches étiquetées, elles désignent les actes de langage correspondants. Le réseau des actes de langage constitue ce qu'on appelle les *conversations pour action*.

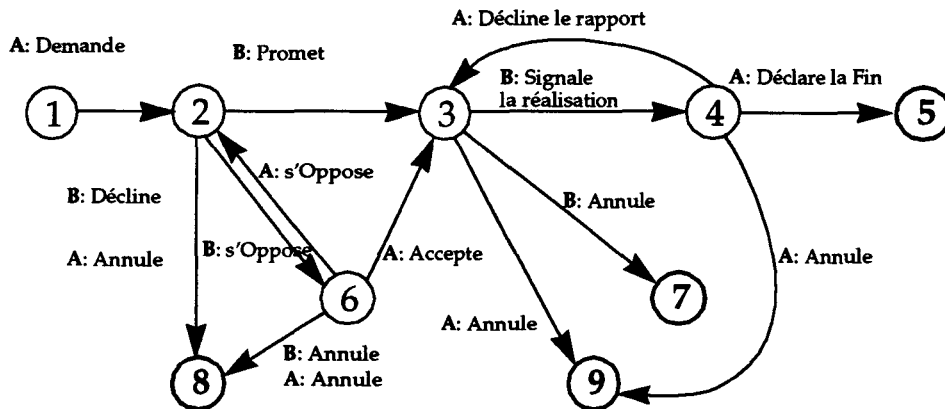


FIGURE 4 La conversation pour action. Ce graphe d'état-transition représente la conversation pour action qui est amorcée par une demande de la part du locuteur A vers l'auditeur B. Les ronds représentent les états de la conversation. Les flèches étiquetées représentent les actes de langage sous-jacents. Les ronds en gras représentent les états qui marquent l'achèvement de la conversation.

Les flèches indiquent les actions pouvant être faites par le locuteur (A) et l'auditeur (B). La requête initiale est émise de A vers B. Elle spécifie des conditions de satisfaction. Pour chaque demande, il existe trois alternatives:

- l'auditeur accepte les conditions de satisfaction (promesse de satisfaction),
- l'auditeur refuse ces conditions,
- l'auditeur demande de négocier une modification ou transformation des conditions (une contre-offre).

Le locuteur peut également retirer sa demande avant toute réponse, ou modifier ses conditions de satisfaction.

Chaque action mène, à son tour, à un état différent avec ses propres possibilités. Dans la situation «normale», B affirme (asserte) au sujet A que les conditions sont satisfaites. Dans ce cas de figure, la conversation évolue vers l'état 4 (c.f. figure 4). Si A déclare la satisfaction, la conversation atteint l'état final 5 qui indique un *accomplissement avec succès*. D'autre part, A peut interpréter la situation différemment, et déclare que les conditions de satisfaction n'ont pas été remplies, renvoyant ainsi la conversation à l'état 3. Dans cet état, chaque partie peut proposer une rectification des conditions. Chaque partie a la possibilité de se dédire et auquel cas, la conversation se termine vers un état d'accomplissement (état 7 ou 9) dans lequel, l'une ou l'autre des deux parties est tenue pour responsable.

2.4 Théorie de la coordination

Le terme de «coordination» est utilisé dans diverses situations et différentes disciplines telles que: la sociologie, les sciences politiques, les sciences de gestion, la théorie des systèmes d'information, l'économie, la linguistique, la psychologie, etc. Chaque spécialiste semble avoir une définition spécifique de la coordination. Toutes ces définitions ne partagent pas encore un cadre commun [Malo90]. Dans cette section, nous allons tenter de définir la coordination en nous référant aux divers travaux de recherche de T. MALONE.

2.4.1 Qu'est ce que la coordination?

Lorsqu'un travail est subdivisé en plusieurs activités, il y a un besoin de coordonner ces activités. Cependant, que signifie exactement la coordination? La définition du dictionnaire est [Malo90]:

- *«La coordination est l'action de travailler ensemble de façon harmonieuse»*

C'est une définition, de fait, à sens très large que nous essayerons, par la suite, de préciser. Mais notons au passage que cette définition contient le mot «harmonieux» qui comprend aussi bien la notion de conflit que celle de la coopération. Car, bien qu'il y ait, dans un groupe, des conflits d'intérêt, on peut juger les résultats des acteurs bons et harmonieux [Malo90].

2.4.2 Autres définitions

Le terme de «coordination» a plusieurs définitions telles que:

- *«La coordination est le nombre des systèmes complexes constitués par les composants» [NSF89].*
- *«Le comportement émergent de plusieurs individus dont les actions sont basées sur des processus de décision complexes» [NSF89].*
- *«Une forme de traitement de l'information dans un système à entités communicantes avec des états d'information distincts» [NSF89].*
- *«L'union des efforts des acteurs communicant de façon indépendante, pour atteindre des objectifs définis mutuellement» [NSF89].*
- *«Les réseaux des actions et engagements humains supportés par l'ordinateur et les technologies de communication» [NSF89].*
- *«La composition d'actions bien définies dans des ensembles bien déterminés» [NSF89].*
- *«L'ensemble des activités requises pour conserver la consistance d'un produit, d'un travail ou pour gérer les dépendances dans le flux du travail: workflow» [Curt89].*
- *«Le traitement de l'information supplémentaire réalisé quand plusieurs personnes connectées ensembles poursuivent leurs buts, qu'un seul acteur poursuivant les même buts ne pourrait réaliser tout seul» [Malo90].*
- *«L'action de gérer les interdépendances entre activités réalisées afin d'atteindre un but» [Malo90].*

Toutes ces définitions illustrent l'aspect complexe de la coordination. Les problèmes fondamentaux de la coordination peuvent se traduire par les questions suivantes:

- Comment peut-on décomposer les buts en actions?
- Comment ces actions peuvent-elles être assignées aux groupes ou aux acteurs individuels?
- Comment les ressources peuvent-elles être allouées entre les acteurs?

- Comment l'information peut-elle être partagée parmi différents acteurs pour aider à réaliser l'ensemble des objectifs?

2.4.3 Composants

T. MALONE propose une décomposition de la coordination en plusieurs composants avec les processus qui leurs sont associés (c.f. tableau 2):

Composant de la coordination	Processus associés
Objectifs ou Buts	Identification des buts
Activités	Planification des buts sous forme d'activités (e.g. décomposition des buts)
Acteurs	Sélection des acteurs, Assignation des activités aux acteurs
Interdépendances	Gestion des interdépendances

TABLEAU 2 *Les composants de la coordination [Malo90]*

MINTZBERG décompose la coordination d'une façon différente de celle de MALONE. Il montre que le noyau de base de toute organisation est formé de *cinq mécanismes* de coordination [Mintz83]. Ces mécanismes désignent les modes qu'utilisent les organisations pour coordonner leurs activités (voir figure 5). Ils sont considérés comme les éléments de base d'une organisation, c'est-à-dire la «substance» qui maintient la consistance d'une organisation:

1. **Ajustement mutuel:** dans le cas d'un ajustement mutuel, le travail est coordonné à travers le processus de communication informelle.
2. **Supervision directe:** en adoptant la supervision comme mécanisme de coordination, une seule et unique personne prend la responsabilité du travail des autres individus. Cette personne donne les instructions et surveille l'exécution du travail.
3. **Standardisation du travail:** on dit que les processus de travail sont standardisés lorsque le contenu du travail est spécifié ou programmé (automatisé).
4. **Standardisation des sorties:** on parle des sorties standardisées si le résultat du travail est spécifié.

5. *Standardisation des techniques et compétences*: les compétences (et les connaissances) sont standardisées si le type de formation requis pour l'accomplissement du travail est spécifié.

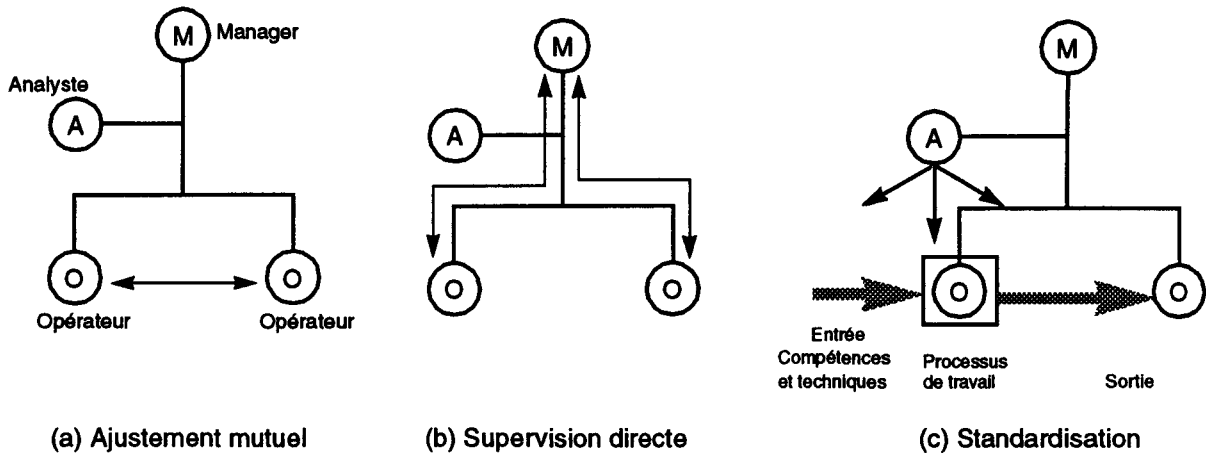


FIGURE 5 Les mécanismes de coordination d'après MINTZBERG.

Lorsque la complexité du travail croît, le mécanisme de coordination utilisé change depuis la *supervision directe* vers la *standardisation des processus de travail*, pour revenir finalement à l'*ajustement mutuel*.

2.4.4 Interdépendance

L'interdépendance joue un rôle clé dans la coordination. En effet, plus l'interdépendance augmente, plus le besoin en coordination dans l'organisation s'amplifie. En réalité, sans l'interdépendance, le besoin en coordination ne peut pas exister. Le tableau 3 illustre les différents types d'interdépendance définis par MALONE [Malo90].

2.4. Théorie de la coordination

Types d'interdépendance	Objet commun	Exemples d'interdépendance dans l'industrie	Exemples de processus de coordination pour la gestion d'interdépendance
condition préalable	Fin d'une activité requise par la suivante	Des pièces doivent être délivrées en temps utile pour être utilisées	Ordonner les activités en déplaçant l'information d'une activité à la suivante
Partage de Ressources	Ressources requises par multiples activités	Installation de deux pièces nécessitant le même outil	Allocation des ressources
Simultanéité	Temps pendant lequel plus d'une activité doivent avoir lieu	Installation de deux pièces en même temps	Synchronisation des activités

TABEAU 3 *Types d'interdépendance.*

2.4.5 Processus sous-jacents aux situations de coordination

Dans le tableau 4, T. MALONE propose quatre couches ou niveaux de processus sous-jacents à la coordination. Chaque couche peut faire appel aux processus se trouvant à un niveau au-dessus.

Niveau de processus	Composants	Exemples de processus génériques
Coordination	but, activités, acteurs, ressources, interdépendances	identifier les buts, ordonner les activités, attribuer les activités aux acteurs, allouer les ressources, synchroniser les activités
Prise de décision en groupe	but, acteurs, alternatives, évaluations, choix	proposer les alternatives, évaluer les alternatives, faire des choix (e.g. par autorité, consensus, ou vote)
Communication	émetteurs, destinataires, messages, langages	définir les langages communs, sélectionner le destinataire (routage), livrer les messages
Perception d'objets communs	acteurs, objets	voir les mêmes objets physiques, accéder aux bases de données partagées

TABEAU 4 *Niveaux de processus sous-jacents à la coordination.*

2.5 L'approche AWA (Action Workflow Approach)

2.5.1 Perspective

L'approche AWA est une approche basée sur la gestion des processus de travail. Elle est décrite comme une méthode systématique pour la conception et la ré-ingénierie des processus de travail dans une organisation [ATI94]. En outre, cette approche permet de caractériser et d'automatiser le workflow. Elle s'inscrit parfaitement dans le domaine du CSCW et du Groupware car elle permet de supporter, dans une certaine mesure, le travail coopératif entre plusieurs personnes. La coordination, la communication et la coopération symbolisent les pierres angulaires de l'activité de groupe [Joos94].

Dans cette section, nous tenterons d'examiner de près l'approche AWA. Pour ce faire, nous nous focaliserons sur les points suivants:

- Que signifie exactement l'approche AWA?
- Quel genre d'utilisation peut-on escompter de cette dernière?
- Quels sont les éventuels inconvénients d'une telle méthode d'analyse?
- Quelle est la procédure de modélisation par l'approche AWA?

2.5.2 Processus de travail

D'après ATI (Action Technologies Inc), la *conversation pour action* peut être représentée par ce que appelle la *boucle* ou le *cycle de workflow* [ATI94]. C'est un workflow élémentaire qui peut être perçu comme une représentation simple du diagramme d'état-transition illustré précédemment. C'est aussi un moyen de schématiser, plus simplement et plus intuitivement, un processus de travail. Par ailleurs, cette boucle met en évidence les phases pendant lesquelles la coordination est réalisée. La figure 6 montre la séquence des actions élémentaires dans un workflow.

Dans chaque boucle, on identifie un *client*, un *fournisseur* et un workflow qui traite d'une action particulière que le fournisseur accepte de réaliser pour la satisfaction du client. Les termes de client et de fournisseur s'appliquent aux individus aussi bien à l'intérieur qu'à l'extérieur d'une organisation.

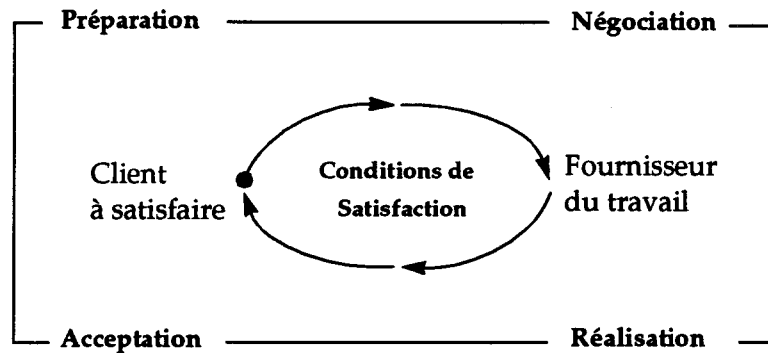


FIGURE 6 La boucle de workflow-action d'après ATI.

Les quatre phases de la boucle fournissent une description du processus de travail à réaliser [Cent93]:

1. *Préparation*: le client émet une demande (ou le fournisseur émet une offre) de réalisation d'une certaine action donnée selon des critères de satisfaction.
2. *Négociation*: les deux parties établissent un accord commun incluant le temps de réalisation pouvant intervenir dans d'autres étapes.
3. *Réalisation*: le fournisseur déclare au client l'accomplissement de l'action.
4. *Acceptation*: le client déclare au fournisseur que la réalisation est satisfaisante.

A chaque étape, on peut avoir des actions supplémentaires telles que des clarifications, des négociations supplémentaires à propos des conditions et des modifications de responsabilités des participants. La structure de workflow telle qu'elle est décrite par la figure 6 se révèle à la fois générale et universelle. En effet, elle est générale car elle a lieu chaque fois qu'il y a coordination entre participants et ce, sans se soucier du travail qu'ils effectuent. Elle est universelle car elle est indépendante de la culture, du langage ou du moyen de communication auquel elle est appliquée.

En reliant les workflow entre eux, on génère, ce que T. WINOGRAD appelle, un *réseau de conversations*. En effet, chaque phase du workflow peut donner lieu à plusieurs sous-workflow (sous-cycles) qui définissent des interactions nécessaires à la réalisation de l'action primaire.

Par exemple, durant la phase de préparation, un client peut initialiser des sous-cycles de workflow qui servent à rassembler les données nécessaires à la formulation de la demande dans le cycle primaire. De façon similaire, un fournisseur peut générer des sous-cycles afin de collecter les informations nécessaires à la formulation de l'offre. Dans la phase de négociation, d'autres sous-cycles peuvent être créés pour supporter la négociation des contraintes de satisfaction entre les deux parties. La phase de réalisation donne lieu typiquement à des sous-cycles de délégation pour la réalisation de l'action en question. Finalement, dans la dernière phase, celle d'acceptation, le client génère des sous-cycles pour l'évaluation du résultat obtenu avant de déclarer satisfaction.

Remarque 1: Les workflow secondaires peuvent être décomposés à leur tour en d'autres sous-cycles. On obtient ainsi un réseau de workflow dans lequel chaque participant joue un rôle bien déterminé.

2.5.3 Comparaison avec les autres approches de gestion du workflow

La plupart des approches de workflow existantes sont structurées autour du domaine d'information [Medi92]. Ces approches sont focalisées sur une classe d'objets d'information, telle que les formulaires ou les images stockées, et définissent le workflow comme une séquence d'actions qui s'exécutent sur ces objets. La structure principale de l'organisation est le «routage» des objets d'information parmi les utilisateurs, et la spécification des actions automatiques réalisables dans ce «routage». D'une certaine façon, cette approche est semblable à celle qui privilégie les processus de matériels où, à titre d'exemple, les pièces passent d'un poste de l'usine à un autre pour le traitement, et où certaines tâches sont réalisées automatiquement par des machines.

Les systèmes workflow, qui adoptent le point de vue des processus d'information et de matériels, se focalisent sur la rapidité de l'exécution d'une étape, au lieu d'encourager les participants à comprendre l'objectif du processus entier.

En outre, contrairement à l'approche AWA, ces systèmes ne gèrent pas le travail qui doit s'effectuer. Ils gèrent simplement le «mouvement» de l'information. Le rôle des participants consiste, en fait, à interagir avec les machines utilisées pour le traitement des informations et des matériels.

Lorsqu'un système workflow est subdivisé en plusieurs étapes pour supporter efficacement le débit du travail, le résultat de cette fragmentation complexifie, pour les participants, la responsabilité de la qualité du travail. La modification ou l'amélioration du système entier demanderait un blocage de tous les processus et des modifications au niveau de toutes les étapes [ATI94].

Par ailleurs, dans ce type de système workflow, l'information spécifique nécessaire, par exemple, à alerter les participants, à guider un acteur dans son travail, etc, n'est pas disponible. L'information présente dans ces systèmes permet aux participants de savoir uniquement ce qui s'est passé, mais pas ce qui doit se passer dans le futur.

Cette approche, orientée processus d'information, est très utile pour la conception des systèmes d'information, mais pas pour les systèmes dédiés à augmenter la satisfaction et la qualité du travail, et répondre aux besoins des clients. Une autre approche est donc nécessaire.

2.5.4 Les inconvénients de l'approche AWA

Nous avons vu que l'approche AWA présente plusieurs avantages. Cependant, elle souffre de quelques faiblesses qui sont énumérées ci-dessous [Joos94]:

1. L'approche AWA permet de décrire la méthode de travail et l'ordre dans lequel les activités sont réalisées de manière statique. Elle ne tient pas compte des contraintes dynamiques qui peuvent provoquer un blocage de l'ensemble du processus de travail. Par exemple, un acteur indisponible à la dernière minute, provoque une suspension de la tâche qui lui incombe. Dans ce cas, les tâches suivantes, qui en dépendent, seront bloquées, et selon l'importance de la tâche suspendue, le système global peut en pâtir sérieusement.
2. La formalisation, la modélisation et l'automatisation des étapes du flux de travail ne sont pas toujours possibles. La conception d'un outil efficace supportant le workflow exige la définition et la spécification des processus formels et informels. Or, dans l'approche AWA, les concepteurs présupposent que toute étape, dans le flux de travail, peut être formalisée, modélisée et automatisée.

3. Dans l'approche AWA, le travail de chaque acteur peut être contrôlé selon les critères de rapidité et d'efficacité dans la réalisation du travail. L'utilisation de ces deux critères peut mettre en péril la bonne ambiance du travail.
4. L'implémentation d'un système de workflow exige beaucoup de changements dans l'organisation du travail. Si cette organisation échoue, pour une raison ou une autre, il est intéressant de pouvoir «revenir en arrière» pour revoir l'implémentation des processus. L'approche AWA, basée sur des boucles de workflow, ne permet pas ce genre de recours. Ceci peut se révéler être un sérieux handicap.

Le dernier point est particulièrement préjudiciable à toute organisation. Lors de l'utilisation d'un système de gestion de workflow dans une unité de travail, ce système devient une pièce-clé de son organisation. Si, pour des raisons données, ce système ne peut plus fonctionner et que le «retour en arrière» dans la modélisation est impossible, l'ensemble de l'organisation de l'unité échoue.

2.6 Outils de Workflow

Pour permettre la gestion des connaissances d'une entreprise, un nouveau type de système est en train de faire son apparition. LotusNotes[®], LinkWorks[®] en sont les principaux exemples. Ces systèmes sont apparus car les outils traditionnels ne sont pas conçus pour permettre la création, la circulation et le suivi d'informations d'entreprise en temps réel, ce qui est la tâche d'une personne dont l'outil de travail est la connaissance.

Nous nous proposons de décrire, dans les sections suivantes, les trois outils de gestion de workflow orientés transactions et qui sont les plus référencés dans le marché du workflow actuellement: LinkWorks[®], TeamLink/TeamRoute[®] (approche DIGITAL) et LotusNotes[®].

2.6.1 Approche DIGITAL

L'architecture de workflow DWA (Digital Workflow Architecture) est une approche spécifique de la société DIGITAL. Cette approche est dédiée à la conception des systèmes de gestion de workflow. Elle se base sur les principes suivants [Digi93]:

1. Le contrôle de flux et la gestion des données doivent être **séparés**. Les systèmes de workflow gèrent les événements complexes, déclenchent les contraintes et les règles d'organisation spécifiées à un niveau supérieur.
2. Les systèmes de workflow doivent être **hétérogènes** et **ouverts**. Une interface ouverte permet à des environnements hétérogènes de participer à la gestion du workflow.
3. Les systèmes de workflow doivent être «**configurables**» pour permettre une grande flexibilité.
4. Les systèmes de workflow doivent être **fiables**. Les composants workflow doivent pouvoir utiliser des transactions distribuées et des services de récupération pour préserver l'intégrité des processus de travail.

2.6.1.1 Vue globale du modèle des systèmes DWA

Comme le montre la figure 7, le principal composant est le contrôleur de flux (Flow Controller).

Le contrôleur a en charge les tâches suivantes:

- gérer les définitions des procédures qui s'appliquent sur le flux,
- gérer les objets de travail actifs (i.e. les flux et les étapes),
- gérer les listes de travail (i.e. file d'attente du travail),
- gérer les données,
- contrôler les ressources et mémoriser une trace des flux d'exécution,
- manipuler les exceptions,
- assurer le respect des règles d'exécution,
- gérer l'historique,
- offrir des services de définition, de création, d'exécution et de contrôle des procédures de travail.

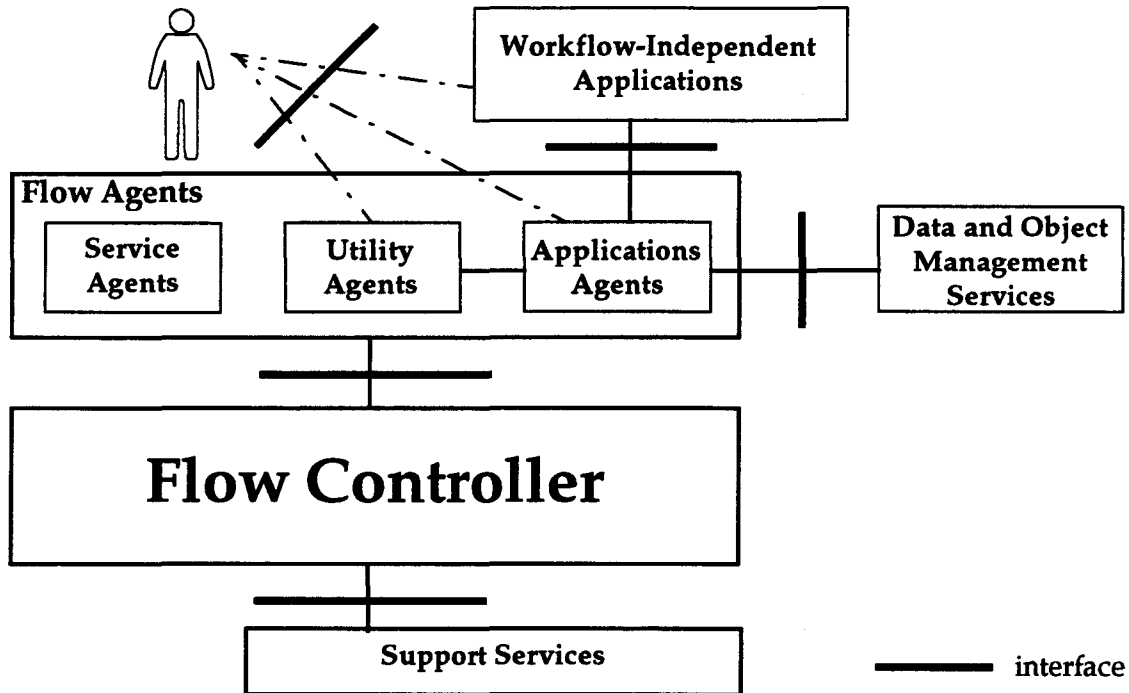


FIGURE 7 Vue globale du modèle des systèmes de workflow selon l'approche DWA (Digital Inc.) [Digi93].

Le modèle DWA classe les agents en trois types distincts:

1. *Agents d'utilité* qui sont des outils de développement tels que l'éditeur, le simulateur, le gestionnaire des listes de travail, etc.
2. *Agents d'application* qui sont des programmes clients qui exécutent des tâches spécifiées par l'utilisateur. Ils sont déclenchés par l'utilisateur pour exécuter des étapes de workflow.
3. *Agents de service* qui permettent au contrôleur de flux de couvrir d'autres domaines et de connecter les contrôleurs de workflow distribués.

2.6.1.2 TeamLinks/TeamRoute

Le système TeamLinks/TeamRoute est une application orientée messagerie. Elle utilise un modèle client-serveur. Le concept de **TeamRoute** est schématisé dans la figure 8.

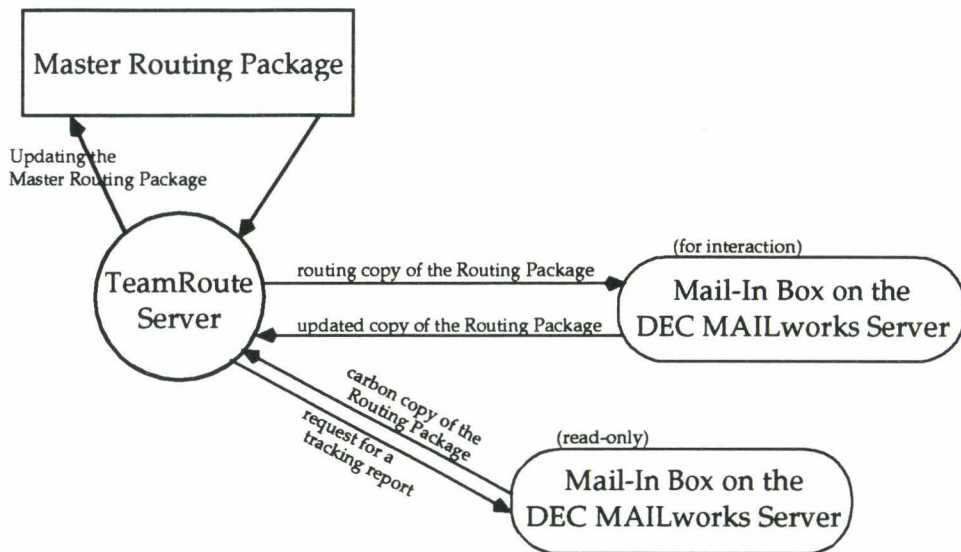


FIGURE 8 Le concept du système TeamRoute (Digital Inc.) [Digi93].

Le client accède au serveur **TeamRoute** pour réaliser le traitement et la gestion des «paquets» d'information. L'utilisateur reçoit les «paquets» comme des messages classiques. Il reçoit une notification lui indiquant l'arrivée des «paquets».

Le modèle **TeamRoute** supporte les interactions d'un groupe de clients hétérogène. Les clients sont connectés à un serveur via un agent de service. Un domaine-serveur comprend tous les agents de service et les clients associés à un serveur particulier.

2.6.1.3 LinkWorks

Le système **LinkWorks** est contrôlé par un moteur de bases de données. Toute l'information qui contrôle le comportement du système est stockée dans des tables.

Dans l'approche **LinkWorks**, le workflow est une liste d'étapes par lesquelles un objet est passé ou doit passer.

Dans **LinkWorks**, il existe deux manières permettant de décrire l'action qui doit s'exécuter à un instant donné:

1. Description par *Modes de workflow*. Un *mode de workflow* est une description de l'état qu'un objet doit avoir avant de passer à l'étape suivante du workflow. L'utilisateur qui

créé un workflow pour un objet, doit sélectionner un *mode* pour chaque étape dans le workflow.

2. Description par des *Remarques*. Dans ce cas, l'utilisateur peut décrire, avec du texte libre, ce qu'il faut faire dans chaque étape.

2.6.2 Approche LotusNotes

LotusNotes peut être considéré comme une extension des outils traditionnels que sont les systèmes de gestion de bases de données et les systèmes de messagerie, avec une architecture spécifiquement conçue pour supporter la gestion et le partage de connaissance [Lotu93]. LotusNotes n'est pas destiné à concurrencer les systèmes de gestion de bases de données ou de messages, mais plutôt à les exploiter et à les compléter.

Pour la plupart des entreprises, il est vital d'élaborer des stratégies fiables et évolutives pour les bases de données et les messageries. De plus, il est important pour toute entreprise dans laquelle la connaissance représente un véritable actif, de développer une stratégie pour la gestion et la diffusion de cette connaissance.

Concrètement, le système de gestion des connaissances d'une entreprise devra présenter les caractéristiques suivantes:

- Etre très ouvert, et capable d'intégrer des documents représentant la connaissance de l'entreprise, y compris les documents générés par les outils informatiques déjà utilisés dans l'entreprise. Les documents créés au jour le jour dans l'entreprise devront bénéficier des mêmes possibilités de partage d'accès et de sécurité que les données transactionnelles des opérations commerciales. Les feuilles de calculs, graphiques, présentations et documents de traitements de texte devront pouvoir être facilement stockés dans la base de données des informations.
- Etre indépendant des topologies, et permettre l'accès à des utilisateurs occasionnels. Cela signifie que le système de gestion de la connaissance d'une entreprise devra être capable, simplement et de manière cohérente, d'intégrer de nombreux sites géographiquement dispersés, ainsi que les ordinateurs à domicile, les portables et les portatifs, ces derniers étant de plus en plus utilisés par les professionnels en déplacement.

- Supporter les modèles de collaboration des utilisateurs (transmission et partage) représentés par les systèmes de messagerie et de bases de données traditionnels. Ainsi, pour n'importe quel élément d'information (feuille de calculs, document de traitement de texte ou note rapide à un autre utilisateur), le système devra pouvoir transmettre l'information à un autre utilisateur d'une manière fiable et performante, et permettre son stockage dans un endroit où les personnes intéressées pourront la consulter rapidement et en toute sécurité. L'information appartiendra alors à la base de données de la connaissance de l'entreprise.

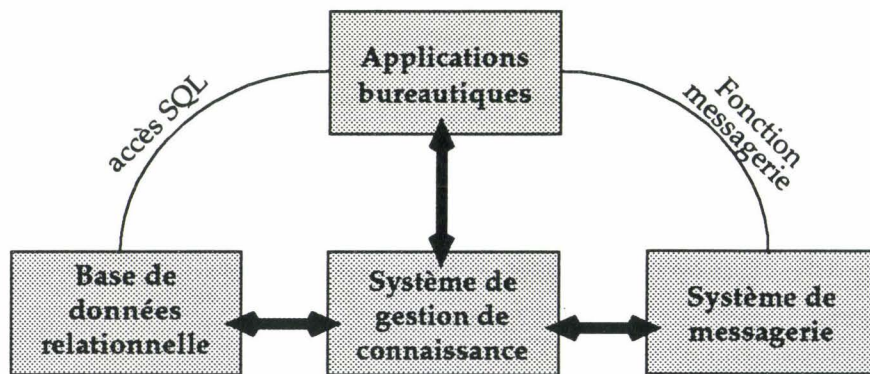


FIGURE 9 Principe de fonctionnement des systèmes conçus selon les principes de «gestion de la connaissance» [Lotu93].

La figure 9 illustre le fonctionnement global des systèmes conçus selon les principes de «gestion de la connaissance». Ils permettent:

- de gérer l'information,
- d'incorporer des applications bureautiques,
- de relier les utilisateurs distants,
- d'intégrer la transmission et le partage,
- de se connecter à des bases de données et des messageries.

2.6.2.1 Objectifs de conception

LotusNotes est une plate-forme qui permet le partage de connaissances et la construction d'applications supportant le travail de groupe. Ce produit a été installé pour la première fois en 1989.

Les objectifs de conception de LotusNotes reposent sur plusieurs principes de base qui ont guidé son développement depuis la première livraison. Comme Notes est composé d'un ensemble de services système et d'une partie applicative, il est nécessaire d'examiner les deux ensembles d'objectifs de conception séparément. Le produit complet, cependant, est une combinaison hautement intégrée de services supportant des caractéristiques et des fonctions applicatives.

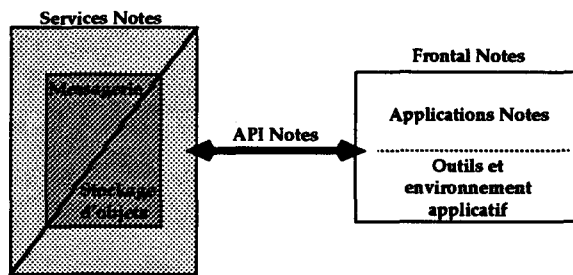


FIGURE 10 Les composants de l'architecture Notes [Lotu93].

Plusieurs technologies centrales utilisées dans Notes, comme le système de sécurité, la réplication, l'architecture multiplate-forme et le stockage des objets (c.f. figure 10), ont été spécifiquement conçues pour répondre aux besoins particuliers des professionnels, en matière de gestion d'informations, dans les entreprises d'aujourd'hui et de demain.

2.7 Conclusion

Dans les domaines impliquant différents processus de travail, les individus sentent le besoin de coordonner leurs activités. L'étude de la coordination dans différents contextes peut contribuer à bien cerner cette notion, à définir un cadre conceptuel de celle-ci et à concevoir des outils de coopération efficaces.

Les outils génériques de groupware fournissent aux utilisateurs une interface commune pour partager un artefact, tel qu'un document sur lequel ils peuvent travailler ensemble. Dans ces systèmes, si la coordination est absente au bas niveau d'interaction (coordination à faible grain), l'activité collective peut devenir impossible. Par contre, au niveau haut de l'activité (à gros grain), la coordination est indispensable pour assurer le bon déroulement des tâches. A ces deux niveaux (à faible et à gros grain), la coordination peut être effective en imposant, volontairement ou involontairement, des contraintes sur les actions de chaque membre du groupe [Proc94].

Nous avons tenté, dans ce chapitre, de décrire le modèle de workflow de WINOGRAD et FLORES, qui est inspiré des travaux de SEARLE sur les actes de langage, ainsi que son intérêt pour la coordination des activités de groupes d'individus au sein de toute organisation. Or ce modèle, bien qu'il présente beaucoup d'avantages en terme de simplicité et d'expressivité, n'est pas très adapté aux organisations où les acteurs sont mobiles et peuvent changer de rôle dynamiquement, car il met particulièrement l'accent sur les interactions humaines aux dépens de l'information manipulée.

Nous verrons par la suite que notre travail de recherche, dans cette thèse, consiste à adapter et étendre ce modèle de manière à ce qu'il soit pertinent pour la description des interactions des individus dans ce type d'organisation. Pour ce faire, nous nous baserons sur une analyse cognitive effectuée au sein des unités médicales.

3.1 Introduction

«*Tout comme une banque ne peut fonctionner effectivement sans des logiciels financiers, il devient de plus en plus difficile d'exercer une médecine moderne sans utiliser les technologies de l'information*». Ces quelques lignes empruntées à SHORTLIFFE [Shor90] témoignent de la progression des techniques et services de l'information dans le monde de la santé. En fait, beaucoup de ces professionnels reconnaissent qu'une grande partie de leur temps et de leurs efforts est consacrée à la gestion de l'information: obtenir et enregistrer des données concernant des patients, consulter des collègues, lire la littérature scientifique, rechercher des informations concernant les services d'un hôpital, d'une clinique ou d'un laboratoire, etc. La liste est longue et on estime volontiers qu'un médecin (généraliste ou spécialiste) passe une heure par jour à rechercher une information qui est dispersée dans les différents supports d'information qu'il est amené à manipuler (VIDAL, annuaires hospitaliers, lettres médicales, résultats de laboratoires, etc.).

Par ailleurs, l'évolution des systèmes de santé et leur régulation nécessitent un suivi accru de l'activité des hôpitaux et des praticiens afin d'assurer un contrôle médicalisé des dépenses de santé, d'optimiser les coûts et de prévoir les dépenses de santé pour les années à venir en fonction des évolutions technologiques et des besoins sanitaires de la population.

Enfin, et ce n'est pas de moindre importance, la patiente -et la population en général- est de plus en plus concernée et attentive à son propre état de santé et à la manière dont les prises en charge et les traitements sont effectués. Le médecin a une «obligation de moyens» pour traiter son patient mais ce dernier voudrait y substituer une «obligation de résultats» ou, tout au moins, s'assurer de la pertinence et de la bonne application des procédures de soins. Ceci se traduit différemment selon les pays et les cultures: par un recours exagéré aux tribunaux aux USA, par une multiplication des consultations de spécialistes en France.

L'information médicale est donc au coeur du fonctionnement de l'activité médicale et dans la régulation du système des soins. Cette information devient d'autant plus nécessaire que la prise en charge des patients s'effectue au cours d'«épisodes de soins» de plus en plus courts, mais pour lesquels une information précise est nécessaire et à l'issue desquels des données nouvelles essentielles sont générées. Le lien entre les divers professionnels, qui sont amenés à prendre en charge un même patient au cours de différents épisodes de soins successifs, apparaît comme un des problèmes essentiels à résoudre dans les prochaines années. L'archivage des données, leur accès sécurisé et confidentiel représentent des challenges informatiques de grande ampleur.

A coté de l'information médicale elle-même, l'informatique a envahi de très nombreux champs médicaux où elle s'est intégrée à l'activité de diagnostics et de soins: imagerie numérisée, interprétation automatique de signaux (électrocardiogrammes, électrocéphalogrammes, etc.) voire d'images (interprétation automatique des mammographies). Un nouveau pas a été franchi ces dernières années avec les nouvelles possibilités offertes par les logiciels de traitements d'images: reconstitutions tridimensionnelles, organes virtuels, simulations d'interventions chirurgicales, et enfin chirurgie assistée par ordinateur où un robot effectue, sous contrôle chirurgical, des interventions extrêmement fines avec une précision qui avoisine les microns.

3.2 Gestion de l'information en médecine

3.2.1 Les systèmes d'information hospitaliers (SIH)

Le terme «*système d'information*» (SI) couvre plusieurs significations. Un tel système peut être vu comme:

- un système de traitement de données moderne bâti autour d'une base de données et utilisant la technologie de l'ordinateur,
- une abstraction d'un système où tous les aspects de représentation symboliques sont ignorés,
- une conception des activités dans une organisation qui supporte et/ou assure la communication dans l'organisation,
- une conception de toutes les données manipulées et activités de communication dans une organisation incluant des sous-systèmes.

Les Systèmes d'Information Hospitaliers (SIH) assurent la communication et la gestion de l'information au sein des institutions médicales. Le développement de ces systèmes a été fortement encouragé aux USA [Hodg92] pour faciliter la gestion financière des structures hospitalières privées et publiques et l'analyse de l'activité médicale et infirmière. Les systèmes de type SIH sont nombreux: les plus connus sont TDS, HELP, PCS, SMS, SICARE. Cependant, tous ces systèmes sont fortement marqués par l'influence anglo-saxonne dans le mode de gestion de l'hôpital. De ce fait, ils s'exportent difficilement et leur utilisation dans les hôpitaux européens est sujette à de nombreuses difficultés de transposition [DeGo94]. C'est pourquoi de nombreux hôpitaux ou structures inter-hospitalières européennes ont développé leurs propres produits: DANNIS (DK), BALIS (ND), CYPHONIE (FR), DIOGENE (Genève-CH), OTALIA (Lille-FR), etc. Ces systèmes ont des fonctions à la fois médicales [DeGo94]:

1. Gestion des données médicales.
2. Gestion des actes médicaux.
3. Demandes/Résultats des actes.
4. Prise de rendez-vous.
5. Evaluation de l'activité.
6. Recherche clinique.

et administratives:

1. Planification des ressources.
2. Comptabilité générale et analytique.

3. Planification financière.
4. Gestion des ressources humaines.

L'objectif d'un SIH est donc double:

- améliorer la qualité des soins en optimisant la circulation de l'information et en assurant une meilleure mise à disposition de cette information,
- optimiser et rationaliser la gestion de l'activité médicale.

Initialement conçus selon une architecture centralisée, les SIH sont de plus en plus développés suivant une architecture distribuée et ouverte (c.f. figure 11) [Sche90] [Dego94].

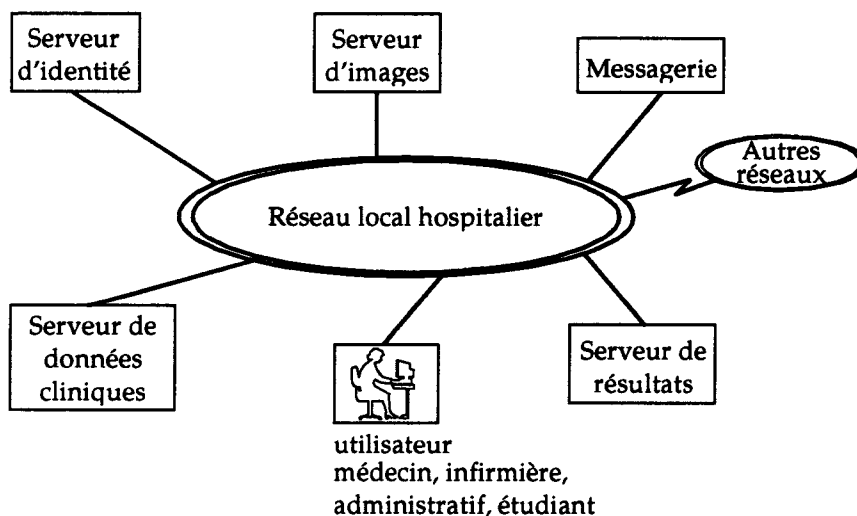


FIGURE 11 Exemple d'une architecture de SIH distribuée basée sur des serveurs.

Le SIH a alors une fonction de communication entre les différents modules fonctionnels. La localisation des ces modules du SIH devient transparente pour l'utilisateur grâce à une architecture distribuée.

A terme, les SIH s'ouvriront au monde extérieur par l'intermédiaire des réseaux de communication. Ainsi, on pourra espérer une meilleure circulation de l'information tant à

l'intérieur de la structure de soins qu'entre ces différentes structures elles-mêmes au sein de ce qui deviendra des «*filières de soins*».

3.2.2 L'intégration

Les systèmes d'information hospitaliers, tels que nous les avons décrits, répondent aux besoins de communication et de gestion médico-administrative des structures de soins, mais ne peuvent satisfaire les demandes spécifiques de nombreux professionnels. Chaque spécialité médicale ou chirurgicale est en effet intéressée par l'utilisation des logiciels spécifiques de traitement de son information et de ses données. Généralement, ces logiciels sont utilisés localement sur un seul site et sur un ou quelques postes de travail. Ce sont au maximum des outils logiciels/matériels utilisés dans un lieu bien délimité voire confiné.

Exemples

- La cardiologie utilise une station d'acquisition des ECG qui en permet la gestion, l'archivage et l'interprétation automatique.
- La radiologie utilise des moyens numérisés d'acquisition d'images (radiologie numérique, CT-Scanner, Imagerie par résonance magnétique, etc.) et des PACS (Picture Archiving and Communication Systems), pour archiver et gérer ses images.
- La médecine nucléaire utilise et interprète des images numériques sur lesquelles elle effectue des traitements complémentaires pour analyser finement le fonctionnement des organes.
- La chirurgie utilise des systèmes de gestion des comptes-rendus opératoires. Certains chirurgiens stéréotaxiques interviennent au travers de robots pilotés par ordinateur.
- L'anesthésie utilise des micro-ordinateurs portables pour la consultation pré-anesthésique en vue de réunir les informations nécessaires à sa pratique.

On dispose donc, dans chaque discipline, d'un ou de plusieurs programmes spécifiques destinés à aider le médecin et/ou l'infirmière dans leurs tâches journalières. Ces outils sont généralement développés de manière indépendante par rapport au système d'information hospitalier.

Il est bien-sûr impossible que tous ces logiciels spécifiques soient développés au sein du seul SIH. L'objectif actuel est donc de les *intégrer*, c'est-à-dire de permettre un *échange sécurisé* d'information entre le SIH et les logiciels spécialisés. Cette intégration nécessite la

mise au point et la généralisation d'outils logiciels intermédiaires (*middleware*) aptes à transmettre, traduire, voire crypter/décrypter des données médicales selon des protocoles totalement sécurisés [ISAR94b].

Cette intégration se réalise généralement au niveau des applications, et non des données [ISAR94b]. L'intégration des interfaces pose des problèmes nouveaux particulièrement intéressants, nécessaires pour l'utilisation par les professionnels. Cette intégration a lieu au travers de traducteurs [ISAR94b] qui assurent les échanges de données et leur mise au format entre, d'une part le SIH, et d'autre part les applications locales médicales (c.f. figure 12).

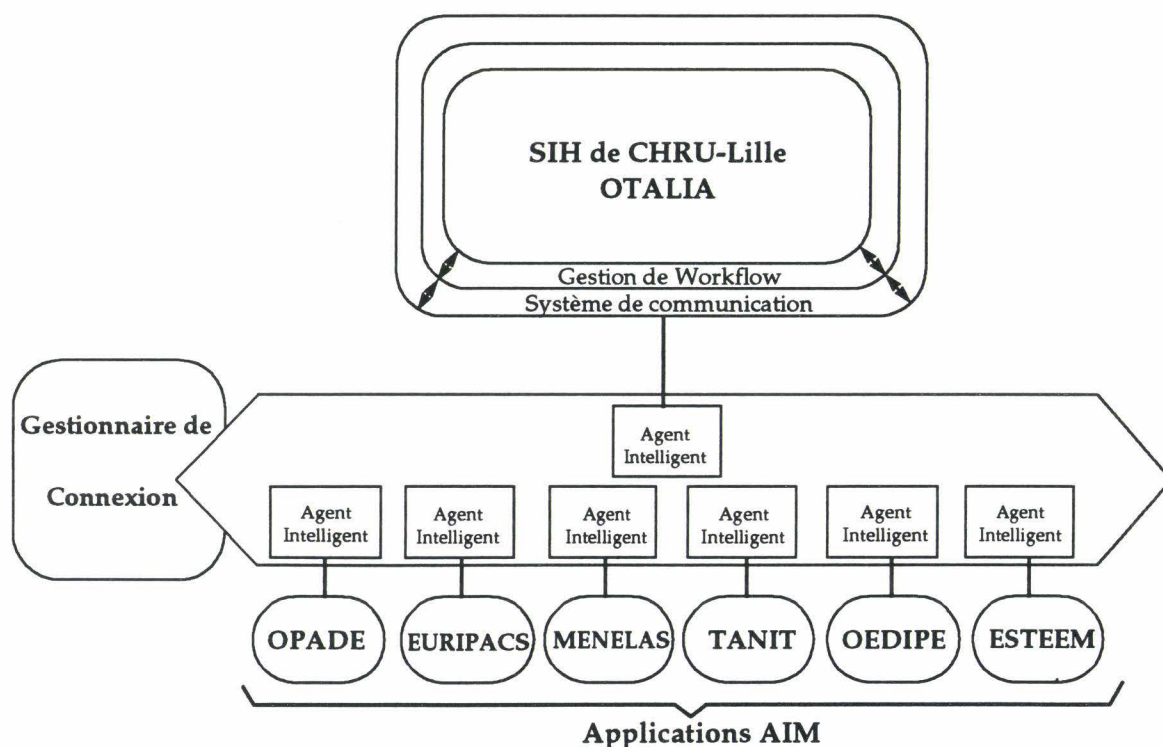


FIGURE 12 L'approche d'intégration ISAR [ISAR94b].

3.2.3 La coopération

Le SIH permet aux personnels hospitaliers de communiquer et d'échanger des données médico-administratives. La réalisation de ces protocoles d'échange s'opère au moyen d'outils logiciels de «*gestion de formulaires*» ou de «*messagerie*». Des outils tels que **LinkWorks** [Digi93], **TeamLink** [Digi93] ou **LotusNotes** [Lotu93] sont expérimentés en milieu hospitalier pour rendre possible la gestion des échanges de données. Sur le plan administratif, cela favorise une meilleure «fluidité» administrative par l'organisation et la répartition des tâches ainsi que pour la coordination des structures de soins. Les nouveaux moyens informatiques sont utilisés pour des tâches essentiellement administratives tels que:

- la transmission de courriers entre les différentes unités médicales et/ou administratives,
- la transmission de formulaires pour:
 - les repos des malades,
 - les mouvements des malades,
 - la commande de produits médico-chirurgicaux,
 - la commande de médicaments,
 - les absences,
 - les demandes de prise en charge de déplacements, missions, voyages, etc.
- la messagerie d'échanges formels ou informels,
- le courrier administratif,
- etc.

Cette coordination informatisée a pour objectif d'optimiser l'utilisation des moyens mis à la disposition des professionnels et d'améliorer, en accélérant, les échanges entre toutes les structures, médicales et administratives, de l'hôpital.

Mais une nouvelle étape mérite d'être franchie.

Maintenant que les SIH existent et sont en mesure d'intégrer des outils spécifiques dédiés à supporter une fonction médicale, il apparaît de plus en plus de moyens de coordination au niveau des unités médicales elles-mêmes. Médecins, infirmières, secrétaires, etc. concourent à une même tâche: assurer au mieux la prise en charge des patients. Mais l'activité médicale est complexe et peu de modèles peuvent rendre compte

de cette complexité [Bowe95] [AIM-TANIT92] [CSIC]. En fait, le médecin et l'infirmière agissent rarement dans l'urgence. Le facteur temps est essentiel: telle action ou telle thérapeutique n'aura d'effet visible et évaluable que dans un certain délai (de plusieurs minutes à plusieurs heures). Ceci nécessite la coordination d'actions désynchronisées entre différents corps de métiers et différentes équipes.

Le lieu où cette coordination s'obscurcit avec le plus d'acuité est l'unité de soins intensifs, lieu où les problèmes de coordination entre soignants et entre équipes peuvent être très critiques.

C'est tout le sujet de ce travail que d'explorer au mieux les besoins, les méthodes et les moyens de coordonner soignants et équipes de soins.

Dans la section suivant, nous décrirons le fonctionnement réel de l'unité de soins intensifs (USI). Nous exposerons, par la suite, ce qu'est le tableau blanc, présent dans la chambre du patient, qui joue le rôle de support d'information partagé et d'interface entre les acteurs. L'information présente sur le tableau blanc n'est pas structurée et est dynamique dans le temps. Nous détaillerons la structure du tableau blanc comme objet central de la chambre du malade: les éléments constitutifs, leur fonctionnement, leur gestion dynamique, etc. Nous proposerons, enfin, une solution permettant de structurer l'information dans le tableau blanc et de supporter la communication informelle entre acteurs, ainsi que la coordination des tâches.

3.3 L'USI et le «Tableau Blanc»

Avant de détailler l'aspect fonctionnel du «*tableau blanc*» dans la chambre du malade, nous nous proposons d'introduire le lecteur dans l'unité de soins intensifs (USI) et d'en décrire le fonctionnement interne, ainsi que le support utilisé pour gérer l'information relative au traitement de chaque patient. Cette description a pour vocation d'expliquer le besoin incontournable:

- d'un support partagé entre les acteurs de soins pour traiter des informations,
- d'un moyen de communication,

- d'un moyen de coordination entre acteurs pour échanger et annoter les informations dynamiques.

3.3.1 Définition d'une unité de soin

«Une unité de soins est une entité physique individualisable dont la fonction est de produire des soins médicaux à visée diagnostique, thérapeutique ou d'évaluation. Il peut s'agir, suivant les cas, d'un service médical ou chirurgical, d'un regroupement de services dans le cadre d'un département ou au contraire d'une unité fonctionnelle à l'intérieur d'un service» [Dego91].

3.3.2 Fonctionnement d'une unité de soin

Dans les USI, on dénombre quatre groupes d'acteurs différents:

1. Les médecins.
2. Les infirmiers (et les sages-femmes en obstétrique).
3. Les aides-soignantes.
4. Les consultants extérieurs au service.

Des subdivisions peuvent exister pour chaque groupe: les médecins seront séparés en praticiens hospitaliers, internes et externes; les infirmiers en surveillants, infirmiers et infirmiers-élèves. Dans chaque groupe, l'ancienneté est prise en considération.

Un service de soins intensifs comporte plusieurs unités regroupant chacune quelques lits. Chaque unité dispose d'un certain nombre de moyens communs pour permettre l'accomplissement des tâches de soins. Par exemple, une unité regroupera cinq lits. Une même paillasse pourra être commune pour la préparation des traitements des patients de cette unité.

La journée commence par les transmissions avec l'équipe de garde. Les infirmières font leurs transmissions aux infirmières, les aides-soignantes aux aides-soignantes, le médecin de garde au médecin. Puis, les médecins effectuent une première visite pour réaliser les modifications thérapeutiques très urgentes.

Dans la matinée, un «staff» réunit les médecins et au moins une infirmière (la surveillante) et une aide-soignante. Le «staff» permet de revoir ou d'établir les grandes lignes thérapeutiques pour chaque malade. C'est également le lieu de discussion et de résolution des problèmes complexes que posent certains malades. Après le «staff», les médecins passent auprès de chaque malade pour établir les prescriptions des prochaines 24 heures. Il s'agit de la deuxième visite effectuée dans la matinée.

Au fur et à mesure de la réalisation des prescriptions, les infirmières préparent les différentes perfusions et réalisent les différentes modifications de prescription. Pour chaque malade, la prescription est consignée sur une feuille de prescription (qui est archivée ultérieurement dans le dossier) et sur le tableau blanc.

L'après-midi, les médecins réalisent les actes médicaux (mise en place des voies veineuses profondes, trachéotomie, etc.); ils accompagnent également les malades en examen à l'extérieur du service (par exemple pour faire un scanner). Le soir a lieu le relais avec l'équipe de garde. Les médecins font des transmissions au médecin de garde, les infirmières aux infirmières de nuit. De nombreuses autres activités s'insèrent dans ce schéma de base: réunions d'enseignements, de bibliographie, etc.

Une bonne répartition des différents acteurs de soins permet d'assurer une cohésion fonctionnelle nécessaire au bon déroulement des activités de cette unité. Un médecin senior supervise une unité. Il est soutenu par un ou deux juniors. Deux ou trois infirmières se répartissent les différents patients. Une unité est toujours caractérisée par la conjonction d'une organisation spatiale et humaine. Les aides-soignantes, quant à elles, se répartissent sur plusieurs unités.

Lorsqu'un membre de l'équipe ne peut assurer une partie de son travail parce qu'il est accaparé par un patient particulier, il est remplacé par un de ses collègues. Certains moments, comme lors des différentes visites, sont des temps de synchronisation de l'équipe. La coopération entre les équipes intervient lors des différentes transmissions: transmissions entre les équipes d'infirmier(e)s et d'aides-soignants qui se succèdent, transmissions entre les médecins de jour et le médecin de garde. La succession des équipes donne à l'activité de soins intensifs un caractère cyclique.

Chaque tâche est attribuée à priori à un médecin ou une infirmière donnés mais elle peut être réalisée par une autre personne selon la répartition de la charge de travail. Le rôle de la personne qui effectue un acte de soin est plus important que cette personne en tant qu'individu.

3.3.3 Spécificités des USI

L'étude psycho-sociologique réalisée en unités de soins intensifs [AIM-TANIT92] a permis de cerner l'importance de la coordination et de la coopération dans l'exercice des tâches de prescription en réanimation. L'activité médicale, particulièrement la réanimation, présente les caractéristiques suivantes:

1. Les contraintes de temps sont fortes et rigides.
2. Les délais d'enchaînement des tâches sont essentiellement définis par la pathologie du malade.
3. Les acteurs de soins ont peu d'influence sur les tâches elles-mêmes.

Dans les USI, l'efficacité de la prise en charge des patients dépend de la rapidité de mise en route et de modification des traitements. Elle dépend aussi de la qualité de la surveillance des paramètres physiologiques (cliniques, para-cliniques, etc.) [AIM-TANIT94]. Ces résultats ne peuvent être obtenus que parce qu'il existe une forte coopération entre les différentes équipes de soins et à l'intérieur de chacune d'elle.

Au sein d'une équipe, la coopération intervient d'une part, entre les personnes ayant des fonctions complémentaires (aides-soignants, infirmier(e)s, médecins, secrétaires) et d'autre part entre, des personnes exerçant la même fonction. La gravité de l'état d'un patient, la réalisation d'examens complémentaires ou d'actes médico-chirurgicaux ou infirmiers, entraînent d'importantes variations dans la charge de travail.

Dans un même service, le cloisonnement entre unités n'est pas étanche et les contraintes de charge de travail ou de personnel peuvent imposer un partage de ressources matérielles (par exemple une chambre de dialyse pour deux unités), et humaines (par exemple, un médecin senior pour deux unités, ou une infirmière s'occupant des patients dans deux unités différentes).

Deux principes régissent l'organisation de travail en USI: la qualité et la proximité.

- **La qualité:** les tâches peuvent être de type médical (e.g. prescription des traitements), infirmier (e.g. dispensation des soins) ou «nursing» (e.g. toilettes, repas, etc.). Les contraintes de temps font que la réalisation d'une tâche est plus importante que son attribution à un membre du personnel. Par exemple, si la prescription à un moment donné ne peut être accomplie par le médecin de l'unité, elle sera effectuée par celui de l'unité voisine. De la même façon, une infirmière disponible pourra être amenée à effectuer les soins à la place de sa collègue accaparée par une autre tâche. Il apparaît donc clairement que la qualité de l'acteur (médecin, infirmière, etc.) prime sur son identité.
- **La proximité:** Dans la mesure du possible, les tâches sont transférées aux collègues qui ont en charge des espaces de travail contigus. De plus, la répartition des tâches en USI n'est pas stricte. Les acteurs de soins ont, par conséquent, la possibilité de modifier cette répartition à tout moment. Cette pratique, devenue courante, rend donc difficile la modélisation de l'activité de soins.

Un facteur supplémentaire s'ajoute à la liste des difficultés: le temps. Les tâches de soins en USI sont très dépendantes du temps. L'évolution des paramètres qui permettent d'apprécier l'état d'un patient est dynamique. Les traitements, ayant par conséquent un caractère variable, doivent être coordonnés tout en tenant compte de leur délai de réalisation. Par ailleurs, les soins doivent être assurés en continu. Aussi, plusieurs équipes se relayent pour assurer cette continuité.

3.3.4 L'information supportée par le tableau blanc en USI

Plusieurs personnes partagent les informations et les tâches concernant un patient donné. Les différents temps de synchronisation (les transmissions du matin et du soir, le «staff», le tour de prescription) ne suffisent pas à assurer un bon partage des informations. Les supports papier sur lesquels sont consignés les prescriptions, les administrations médicamenteuses, les soins et la surveillance sont difficilement modifiables et ne permettent pas de mettre en évidence une information capitale. C'est pourquoi, de nombreux services ont pris l'habitude d'utiliser un «tableau blanc» pour la gestion quotidienne des événements. Sur ce tableau blanc sont consignées et mises à jour les informations concernant:

- l'état du patient, les principaux antécédents, le diagnostic du problème actuel,
- les consignes de «nursing» et d'alimentation (à jeun, ne pas asseoir le patient, etc.),
- les examens et rendez-vous prévus (scanner le xx/xx/xx),
- les paramètres à surveiller et la prescription.

Ce tableau est donc un espace d'information partagé. Il ne contient pas toutes les informations, mais seulement celles utiles aux 24h en cours. Une partie des éléments du tableau font référence au dossier du patient. Cet espace commun est à la fois un espace d'information (résumé des informations les plus importantes et qui sont d'actualité) et un espace de communication. Les éléments modifiés seront vus par tous. Ils peuvent être nominatifs, s'adresser à l'ensemble ou à un sous-ensemble d'acteurs de soins.

Actuellement, le tableau blanc, utilisé en USI comme support d'information partagé, se révèle inefficace. Les principales causes sont:

- la structuration semi-anarchique de l'information qu'il contient,
- la vue restreinte de l'information qui couvre uniquement les 24 heures de traitement.

Par conséquent, la recherche de l'information, à un instant donné, est une tâche délicate et fastidieuse.

En outre, la communication asynchrone entre acteurs se fait au moyen des annotations sur le tableau blanc. Elle est donc «noyée» dans l'information propre au patient. Ceci peut donc perturber la coordination des tâches de soins et conduire à des incidents préjudiciables au bon fonctionnement de l'USI. La qualité des activités de soins en USI est donc perfectible.

La question soulevée est donc de savoir quelle solution on doit adopter en USI pour structurer l'information utile sur le tableau blanc et bien séparer la communication de la coordination des tâches.

3.3.5 La solution des «post-it»

Au sein des organisations, les individus utilisent les «*post-it jaunes*» pour communiquer de façon asynchrone. Ce concept facile, simple et efficace (dans une certaine mesure), a bouleversé le quotidien. La preuve en est évidente, car les «post-it» sont présents partout!

Dans les USI, les acteurs de soins pourraient utiliser ce moyen non seulement pour communiquer en mode asynchrone, mais en plus, pour coordonner leurs tâches et échanger l'information pertinente qui évolue dynamiquement au fil du cycle de traitement des patients.

Le tableau blanc qui sert de support de partage d'information et d'interface entre les participants se révèle insuffisant pour deux raisons principales:

- d'abord, à cause de la quantité importante de l'information véhiculée, et ce, quelque soit la taille du tableau,
- ensuite, à cause de la structuration *semi-anarchique* de l'information présente sur le tableau. Ceci implique, de toute évidence, que la recherche d'information et notamment des données pertinentes s'avère très lourde, coûteuse, ennuyeuse et surtout matière à de nombreuses erreurs de lecture et d'interprétation.

Il apparaît donc un besoin naturel et urgent d'avoir un moyen complémentaire qui permet de supporter cette information dynamique et d'assurer la communication asynchrone et la coordination entre les participants d'une USI.

Le moyen que nous proposons est le «post-it». Il permet aux médecins et aux infirmières de s'échanger des messages, reporter des annotations, des remarques et commentaires sur l'état de santé du malade, sur les traitements prescrits, etc. Une infirmière peut demander, au moyen d'un «post-it», une clarification à un médecin prescripteur absent du service ou simplement occupé. Un médecin de garde peut annoter sur le tableau blanc, au moyen d'un «post-it», des informations explicatives concernant une prescription qu'il a supprimée, à l'intention des membres de l'équipe du jour. Par ailleurs, un «post-it» peut comporter tout simplement des messages personnels, en l'attachant à un endroit que le destinataire est sensé occuper (par exemple, sur son bureau, sur l'écran de son ordinateur, etc).

La vocation du «post-it», par définition, est de souligner une information pertinente vis-à-vis de l'auteur du «post-it» et du destinataire. Il ne doit donc pas contenir le contexte global ou le détail sur le sujet principal, mais il doit le référencer. Pour ce faire, les acteurs de soins attachent les «post-it» sur le tableau blanc, à des endroits différents, quand il s'agit d'une information concernant le patient et son état de santé. Le tableau blanc est donc un

lieu de rencontre «asynchrone», où tous les acteurs viennent chercher ou mettre à jour les informations liées aux traitements prescrits pour un patient.

Cette solution à base de «post-it» présente l'avantage de:

- structurer, dans une certaine mesure, l'information,
- faciliter la recherche de l'information,
- coordonner les tâches entre acteurs en spécifiant les tâches accomplies et celles non encore réalisées,
- assurer une communication asynchrone entre les acteurs de soins.

Limitations

Cependant, un mécanisme de coordination et de communication aussi informel que le «post-it» a ses limites. Il demande une gestion stricte de la part des acteurs, rôle qui n'est pas facile à assumer. Tous les acteurs sont tenus de mettre à jour les «post-it» qu'ils ont attachés sur le tableau blanc et ceux qu'ils ont reçus. De plus, certains «post-it» sont destinés à plusieurs acteurs. Il faut donc s'assurer que tous les destinataires du message en ont eu connaissance avant de le mettre à jour. Ceci complique la gestion des «post-it».

D'autre part, si le tableau blanc est rempli de «post-it», sa fonction de coordination perd quelque peu son sens. Les acteurs laisseraient vraisemblablement tomber cette solution fastidieuse et coûteuse en temps.

Force est de constater qu'un moyen de gestion automatique de ces «post-it» s'avérerait beaucoup plus efficace et plus sûr. L'introduction de l'informatique dans la gestion de l'information et des «post-it» dans le tableau blanc est donc indispensable.

3.4 Conclusion

La santé est l'un des besoins humains les plus fondamentaux. Par conséquence, elle est l'une des premières occupations de toute société [Blum90]. Cependant, l'application de l'informatique à la médecine est relativement récente, et les systèmes informatiques dédiés à la médecine ne cessent de se multiplier d'année en année.

La technologie des systèmes d'information hospitaliers continuera à se diversifier selon les institutions. Les grandes structures hospitalières, qui supportent les activités tertiaires,

possèdent des dispositions financières complexes avec d'autres institutions universitaires, hospitalières, gouvernementales, etc. Pour répondre à leurs besoins de gestion, la majorité de ces structures doivent concevoir, développer et maintenir leur propre système d'information. Par contre, les petites et moyennes structures hospitalières sont très dépendantes des applications commerciales développées par des vendeurs informatiques. Ceci implique, de toute évidence, que les SIH soient très variés et ne répondent à aucun standard.

La croissance des demandes cliniques et administratives, avec celle des technologies, devrait encourager vivement le développement des systèmes d'information dans les années à venir. Cependant, à cause du facteur économique, en particulier dans le domaine de la santé, les hôpitaux ne jouissent plus du même degré de liberté pour les expérimentations des SIH. De plus, la recherche scientifique n'est plus aussi vaste qu'il y a dix ans [Shor90]. Il faut donc espérer que ces hôpitaux s'intéresseront à répondre aux besoins pratiques à court-terme, et utiliseront les résultats de la recherche scientifique à long-terme.

Par ailleurs, l'informatisation des unités de soins pose des problèmes complexes de modélisation du système d'information et de stratégie de mise en place [DeGo90]. Les choix effectués au niveau d'un établissement peuvent être conditionnés par la demande des utilisateurs, l'infrastructure informatique déjà en place et/ou les contraintes financières. Ceci pose de sérieux problèmes d'intégration dans les SIH.

L'informatisation des unités de soins est contemporaine par rapport à la mise en place des premiers systèmes d'information hospitaliers. Elle ne peut d'ailleurs en être dissociée.

Nous avons pu voir les unités de soins intensifs représentent une structure organisationnelle complexe, dont la hiérarchie est peu profonde et dont la coopération est principalement verticale. La contrainte de mobilité, imposée aux acteurs, est l'un des facteurs caractéristiques des unités de soins. Cette contrainte, à la fois spatiale et temporelle, fait que le fonctionnement dans les unités impose un minimum d'organisation et un maximum de coordination des activités de soins entre les intervenants. Il n'y a coopération effective que s'il y a une coordination efficace [Malo90].

3.4. Conclusion

Partant de ce principe, nous nous devons de voir de près, quels sont les vrais problèmes engendrés dans les unités de soins actuellement, et pourquoi les systèmes informatiques classiques (mono-utilisateurs) échouent dans la réponse à ces problèmes.

PARTIE II

La Proposition PLACO

Dans cette partie, nous nous intéresserons à la description de la proposition d'une solution que nous avons nommée PLACO.

Cette partie se décomposera en quatre chapitres traitant respectivement:

1. des problèmes de fonctionnement rencontrés dans les unités de travail. Il décrira l'étude socio-psychologique effectuée au sein d'une unité de soins intensifs afin de recenser les incidents et accidents qui surviennent et perturbent le fonctionnement normal des unités de travail.
2. du modèle de Workflow que nous proposons pour la gestion des interactions humaines dans les unités de travail. Ce modèle se base principalement sur:
 - la théorie des actes de langage de J. R. SEARLE,
 - la théorie de conversation de T. WINOGRAD et F. FLORES,
 - la théorie de coordination de T. MALONE.

Le modèle est un cycle de Workflow principal décrivant les actions et les rôles qui entrent en jeu dans les unités de travail. Il inclut également les activités dynamiques, inhérentes aux organisations à faible couplage, en permettant aux acteurs d'entreprendre des conversations de délégation et de clarification de leurs tâches.

3. de l'application du modèle à la planification coopérative en unités de soins intensifs (USI) en se basant sur l'utilisation d'un support partagé d'information qui joue également le rôle de support de coordination entre les acteurs de soins.
-

4. de la réalisation d'un prototype en Smalltalk et de la validation du modèle. Le prototype est destiné à démontrer les intérêts et les bénéfices que peut apporter l'introduction d'un outil de coordination au sein des organisations où les acteurs et les tâches qu'ils sont amenés à remplir, sont faiblement couplés. Une évaluation des différentes fonctions du prototype, par un médecin spécialisé dans l'ergonomie du travail, sera faite afin de nous permettre de mieux estimer les réels bénéfices escomptés.

*Coopération et
Coordination:
Application à l'USI*

De nombreuses difficultés surviennent lors de la réalisation d'une tâche complexe impliquant de nombreux partenaires (ce qui est le cas des unités de soins intensifs).

Une de nos hypothèses de travail, issue des résultats préalables des projets européens ICSIC [] et TANIT [AIM-TANIT92] [AIM-TANIT93] [AIM-TANIT94], est que les erreurs, les «breakdowns», sont dûes à des défauts de coordination des différents acteurs. A cette fin, nous avons mené, dans les unités concernées, une observation destinée, dans un premier temps, à recenser les «breakdowns», dans un second temps, à expliciter les causes et les effets sur l'organisation de l'unité, ceci dans le but de bâtir un outil informatique adapté à la résolution des problèmes ainsi mis en évidence.

4.1 Analyse Socio-Psychologique des «breakdowns»

4.1.1 Définition

Dans notre étude, le terme «breakdown» signifie un «écart» par rapport au fonctionnement normal d'une unité suite à une perturbation ou un incident. Il s'ensuit qu'un «breakdown» n'induit pas forcément une situation néfaste et préjudiciable au malade mais simplement une perturbation dans l'organisation de l'unité de travail.

Cette définition est généralisable à toute unité de travail, nous l'appliquons, pour l'exemple, à une unité de soins intensifs pour laquelle nous développons l'application PLACO (PLAnification COopérative).

4.1.2 Hypothèses de l'étude

En considérant l'hypothèse selon laquelle les «breakdowns» découlent des déficiences et des troubles de fonctionnement des interactions entre individus, on peut penser, à priori, qu'un système de type «groupware» pourrait venir en aide aux acteurs de soins afin de combler ces carences de coordination d'origine humaine. Cependant, avant de s'engager dans des considérations d'ordre informatique, nous avons besoin, au préalable, de:

- connaître précisément les caractéristiques de ces «breakdowns» dans les unités de travail concernées,
- analyser les causes réelles et potentielles des «breakdowns»,
- caractériser précisément l'importance des divers «breakdowns» en termes de:
 - conséquences pour l'organisation de l'unité et pour les résultats de l'action entreprise,
 - fréquence, du fait que certains «breakdowns» sont quotidiens, d'autres sont exceptionnels.

Tels sont les objectifs de cette étude. Cette dernière nous permettra d'affirmer ou infirmer qu'un système «groupware» supportant la coordination est la solution recommandée pour éviter les «breakdowns».

4.1.3 Méthodologie de l'étude

La première étape, avant toute analyse, consiste à recueillir les données relatives aux «breakdowns» que l'on est en mesure d'observer dans l'unité de travail concernée.

A cet effet, nous avons organisé des réunions dans une unité médicale de soins intensifs de l'hôpital CHRU de Lille, auxquelles plusieurs personnes de profil différent ont participé tels que:

- des médecins (internes et chefs de service),
- des infirmières et des infirmières-chefs,
- des aides-soignantes.

Les conduites de réunions étaient focalisées sur deux objectifs principaux:

- initier des participants aux notions de «collaboration» et de «co-décision», et définir les informations pertinentes (l'objectif de la première réunion),
- rapporter et reporter les données liées aux informations pertinentes (l'objectif des réunions suivantes).

Remarque 1: Les réunions s'avéraient très fructueuses du fait qu'elles permettaient de valider instantanément les données collectées.

Au total, quatre rencontres ont eu lieu dans l'unité de soins intensifs et d'urgence en collaboration avec le département d'anesthésie de l'hôpital Claude Huriez du CHRU-Lille.

Les rencontres étaient programmées deux fois par mois et groupaient les personnes suivantes:

- un psychologue chargé de l'étude,
- un médecin,
- trois infirmières,
- deux aides-soignantes,
- quelques observateurs.

Remarque 2: Occasionnellement, d'autres personnes de profil similaire ou différent ont participé aux réunions, tels qu'un deuxième médecin, une infirmière-chef, un physiothérapeute, etc.

4.1.4 Analyse des résultats

Les participants ont été amenés à reporter tous les incidents, en terme de coopération et de coordination, avec une description des situations de conflits, que l'on peut observer dans leur service.

L'analyse des processus de «breakdown» s'est basée sur trois critères résumés dans le tableau 6.

Les rapports de réunions ont permis de souligner quelques éléments constants et communs à tous les scénarii. Pour chaque scénario, nous avons identifié six facteurs principaux:

- les acteurs médicaux impliqués dans le scénario: victimes, témoins ou accusés,
- le fait qu'un document écrit avait été ou aurait pu être utilisé dans cette situation,
- le fait qu'il y avait une transmission orale (directement ou par téléphone) des données entre acteurs,
- les causes attribuées par les acteurs des «breakdowns». Ces causes sont explicitées par les témoins de la session. Elles peuvent être sûres ou probables,
- les conséquences qui découlent de la mauvaise coopération au sein de l'unité. Ces conséquences peuvent être préjudiciables à la santé du patient ou simplement, à l'organisation des soins,
- la fréquences des «breakdowns» observés.

Élément d'analyse	Détail
Rôle des personnes impliquées	témoins, victimes ou accusés (coupables ou non).
Causes des «breakdowns»	réelle, probable ou possible.
Conséquences pour	l'organisation de l'unité de travail.

TABEAU 5 *Principaux éléments de la méthodologie utilisée pour l'analyse des rapports de réunions.*

Sept contextes ont été retenus pour un total de vingt situations. Ces contextes sont favorables à la survenue des incidents. Ils sont récapitulés dans le tableau 7.

Contextes	Nombre de situations observées	Fréquence estimée
1. Situation normale (relaxation de l'attention des acteurs)	4	++
2. Modification d'une prescription	6	++++
3. Planification d'un examen	4	++++
4. Prescription d'un traitement exceptionnel	1	+
5. Problème durant le processus de décision	3	++
6. Séquence de prescriptions	1	+++
7. Interventions extérieures	1	+

TABEAU 6 Nombre de situations reportées pour chaque contexte. L'importance du contexte pour un acteur médical est exprimée par des signes "+".

En analysant ces contextes, nous avons pu caractériser et déterminer les causes et les conséquences des «breakdowns»:

4.1.4.1 Relâchement de l'attention du personnel en situation normale

Lors des situations monotones et routinières, un manque de vigilance par relâchement de l'attention des soignants donne lieu à des rapports incomplets, des tâches triviales non-accomplies, une moindre qualité de la surveillance, etc. Si un événement grave ou imprévu survient alors, incidents ou accidents peuvent survenir avec des conséquences majeures.

4.1.4.2 Modification d'une prescription sans explication et sans justification de la décision

La personne supposée exécuter l'action de modification:

- n'est pas informée au préalable de l'action elle-même, ou ne reporte pas ladite action,
- ou, reporte l'action de modification mais sans explication suffisante, sans indication de l'identité de la personne qui doit exécuter cette tâche ni de la raison de cette modification,
- ou, n'a pas l'information complète et nécessaire pour l'accomplissement de l'action.

Ces incidents résultent finalement d'une mauvaise transmission (i.e. mauvaise coordination) entre les acteurs de l'équipe thérapeutique. Ces situations induisent une

perte de temps pour la vérification, et probablement un ajournement ou une annulation de l'action.

4.1.4.3 Décisions partielles non mises-à-jour

Au sein des unités de travail, les décisions incomplètes et inadaptées sont parfois sources d'incidents. Les principales causes sont les suivantes:

- les participants au processus de décision ne sont pas forcément tous présents en même temps, ce qui induit une prise de décision approximative.
- le manque de pertinence dans l'information utile, la surcharge d'informations, les délais d'attente, la qualité insuffisante des données, etc.

Comme conséquence immédiate, les décisions prises sont vagues et imprécises et exigent une validation ou une confirmation qui, probablement, n'aura jamais lieu.

4.1.4.4 Changement non reporté des règles de travail

Que ce soit une modification des règles de dispensation de soins, ou des règles de demandes et transmissions des examens, prélèvements ou autre, la multiplication des intervenants induit des erreurs et des malentendus lors de l'interprétation des ordres.

4.1.4.5 Confusion dans les étapes du travail

- Certaines décisions peuvent être exécutées sans avoir été explicitées dans le plan du travail. Dans ce cas, l'action ne paraît avoir aucune référence à une planification, ni explication, ni justification.
- D'autres décisions planifiées peuvent être exécutées mais sans être reportées explicitement, ce qui entraîne des erreurs souvent irrécupérables.

4.1.4.6 Complexité des processus de soins nécessitant une longue chaîne de participants

Planifier et organiser un examen médical qui s'effectuera à l'extérieur du service nécessite généralement des étapes multiples (e.g. des séquences complexes de médicaments). Ceci peut engendrer une annulation ou un ajournement de l'examen en question, ou la perte de l'information et la confusion des étapes thérapeutiques.

4.1.4.7 Délégation impropre

L'information appropriée n'est pas transmise convenablement aux infirmières et aux médecins, lors des roulements. Ceci induit une mauvaise connaissance du processus de soins appliqué à un patient donné. Les nouveaux arrivants n'ont pas l'information pertinente pour avoir une vue globale de ce qui s'est passé au préalable afin d'agir en conséquence. Les données transférées restent généralement à l'état brut et ne sont donc pas exploitables.

4.1.4.8 Inadéquation entre ce qui est dit et ce qui est fait

Les problèmes sont traités à différents niveaux d'abstraction. Les décisions sont prises au niveau conceptuel sans aucune vérification des ressources ni des conditions du patient. Des ordres «ouverts» sont émis alors sans indication explicite des conditions d'action.

4.1.5 Conclusion de cette analyse

Durant cette expérience, nous avons pu déterminer certaines situations particulièrement propices aux erreurs et incidents. L'analyse de ces situations nous a permis de conclure que la plupart des incidents, lors de la prise en charge médicale d'un patient, affectent principalement la transmission des données et le transfert de l'information¹ entre les acteurs de soins, pendant le passage d'une action à une autre, principalement les actions de prescription et de dispensation.

Cette analyse a mis en évidence quelques éléments essentiels qui mériteront une investigation sérieuse. En effet, nous avons pu montrer que certains contextes semblent être particulièrement propices à la survenue des «breakdowns». Les causes principales sont récapitulées comme suit:

- faiblesse des modes de comportement cognitif des individus et de leur contrôle (c.f. 4.1.4.1),
- faible communication de la simple information,
- difficultés dans la synchronisation de tous les aspects de la prise de décision(c.f. 4.1.4.3),

1. Le terme «information» peut signifier des données cliniques, des rapports médicaux, des modifications de soins, etc.

- différences, déformations et déviations dans l'interprétation des décisions et des données,
- faiblesse des mécanismes de régulation et de contrôle pour régénérer des décisions, garantir l'enchaînement des étapes de travail et assurer le transfert de l'information et des responsabilités.

On voit donc, à l'issue de cette phase, que la majorité des «breakdowns» sont liés à une mauvaise distribution de l'information entre les différents acteurs: l'information n'est pas écrite, ou disponible, ou transmise, ou n'est pas évaluée, ce qui génère des incohérences dans le déroulement du travail.

Cette première phase s'est avérée insuffisante pour caractériser toutes les situations évoquées, ou pour diagnostiquer toutes les sources réelles des «breakdowns». Cependant, cette démarche s'est révélée fructueuse puisque les résultats obtenus confirment la présence des «breakdowns» dans les unités de travail, et que ces erreurs affectent principalement la transmission des données entre acteurs de soins. Le vrai problème incontournable est l'inadéquation entre la prescription d'un traitement et sa dispensation par les infirmières ou les aides-soignantes.

4.2 Choix du thème d'application

Nous avons choisi la *planification thérapeutique* comme thème de modélisation et d'application du travail coopératif et du concept du CSCW au sein des unités de soins intensifs (USI). Les motivations de ce choix sont les suivantes:

1. Notre analyse socio-psychologique des «breakdowns» a montré que la prescription-dispensation des soins en USI est l'une des tâches les plus importantes et les plus complexes où les incidents surgissent en grand nombre. La majorité des «breakdowns» recensés concerne cette tâche, que ce soit la prescription d'une thérapeutique, d'un examen ou d'une conduite à tenir.

La planification thérapeutique est ainsi un domaine où la collaboration entre individus est, à la fois, étroite et discontinue, imposant ainsi une coopération asynchrone. C'est donc

un champ qui se prête bien au test des nouvelles idées et concepts du CSCW et du travail coopératif.

2. Par ailleurs, la démarche-utilisateur [Ota192], réalisée au sein du CHRU de Lille pour le SIH¹, a permis de soulever plusieurs difficultés dans le domaine de la planification thérapeutique. Cette activité nécessite une parfaite coordination des activités entre plusieurs individus de culture sociale différente:

- les médecins pour la prise de décisions, le choix des thérapeutiques et les responsabilités qui en découlent,
- les infirmières pour la dispensation des soins, la mise à jour des traitements, la gestion des conflits et la collecte des données,
- les aides-soignantes pour les tâches complémentaires (alimentation, surveillance, etc.),
- les autres intervenants tels que les physiothérapeutes, nutritionnistes, les ouvriers hospitaliers, etc.

Remarque 3: Plus le nombre de participants est élevé, plus les irrégularités du fonctionnement entre les personnes ayant en charge le patient se multiplient.

4.3 Proposition de la solution PLACO: une Architecture Coopérative Distribuée

L'analyse du travail dans une unité médicale nous a permis de démontrer rapidement la nature coopérative du travail qui s'y effectue. L'analyse fonctionnelle des tâches réalisées dans une USI identifie ainsi le rôle du médecin, celui de l'infirmière, de l'aide-soignante ou de l'hôtesse, de manière séparée et indépendante. Cette vision retient sur la mise au point des outils informatiques et des logiciels qui leur sont destinés: les notions de personnalisation, d'adaptation au contexte de travail ont concouru à séparer les activités de chacun en leur donnant un accès différencié aux informations.

Etant donné la complexité de la planification thérapeutique, il va sans dire qu'un système informatique de planification coopérative pourrait être une des réponses aux «breakdowns» recensés et, à terme, améliorer la qualité des soins.

1. le Système d'Information Hospitalier *OTALIA* réalisée à Lille.

Afin de mettre en évidence la place de la coopération et de la coordination des tâches lors d'une prise en charge médicale, nous nous sommes intéressés aux architectures distribuées favorisant le traitement des malades hospitalisés en USI et réanimation et permettant de supporter la coordination et la coopération en USI.

L'analyse que nous avons effectuée nous a amené donc à proposer une architecture logicielle distribuée.

4.3.1 Spécifications

Pour ce faire, nous proposons une plate-forme coopérative appelée **PLACO**. Celle-ci est bâtie de la manière suivante:

1. Un tableau d'objets partagé sur lequel les acteurs peuvent intervenir pour:

- créer, modifier, détruire les objets, c'est-à-dire:
 - prescrire un traitement,
 - modifier une prescription,
 - arrêter un traitement,
 - signaler qu'un traitement a été administré.
- lire des informations ou des messages:
 - des autres membres de l'équipe,
 - d'autres sources (laboratoires, comptes-rendus, etc.),
 - de bases de connaissances,
 - du processus de contrôle.

2. Un processus de contrôle du flux de travail, destiné à contrôler les procédures mises en oeuvre et à vérifier que les informations nécessaires sont présentes avant de passer à l'étape suivante (exemples: on ne peut donner un médicament non prescrit; l'arrêt d'une ventilation doit faire l'objet d'un commentaire; la prescription d'un protocole doit être référencée, etc.). Ce processus de contrôle met donc en oeuvre le circuit de fonctionnement du flux de travail qui pourra se faire par envoi de messages entre les membres de l'équipe. Certains messages seront émis automatiquement lorsque la procédure est explicitée.

3. L'accès à des bases de connaissances et des bases de données pour aider, documenter et contrôler les prescriptions. Les objectifs recherchés sont:

- la vérification de la posologie,
- la recherche des interactions médicamenteuses,
- l'aide à la dispensation,
- le contrôle de l'existence d'allergies ou d'intolérances connues,
- etc.

Comme le montre la figure 13, les éléments constitutifs de l'architecture que nous proposons sont détaillés comme suit:

1. **Tableau A:** on dispose d'un tableau A, contenant des objets (objets d'information) manipulés, à la fois, par les acteurs et par les processus de test (de cohérence, de protection, etc...) à travers les bases de connaissances. Autour de ce dernier, on trouve les tableaux de contrôle B et C.
2. **Tableau B:** spécialisé dans le contrôle de cohérence des objets du tableau A.

Par exemple, un médecin prescrit un médicament qui ne se trouve pas dans la Base de Données des médicaments, chose qui doit logiquement générer une erreur. Ou encore, un médecin substitue à un médicament son équivalent et par suite, omet de changer sa forme (sachet, sirop, etc.), ou indique une posologie inadaptée.

Le tableau en question permet également de protéger des données par verrouillage de ces dernières. On peut imaginer beaucoup de raisons pour le faire, telles que l'interdiction de modification ou de consultation de certains objets à une certaine classe d'acteurs: infirmières de nuit par exemple. Pour cela, les acteurs doivent être hiérarchisés.

3. **Le tableau C** est spécialisé, quant à lui, dans la gestion des interactions entre acteurs autour du tableau A. Il assure les fonctions principales suivantes:

- Contrôler les changements effectués par les acteurs sur les objets d'information, établir une sorte de coopération entre utilisateurs . La gestion du flux de travail est donc contrôlée par ce tableau.
- Gérer les différents messages véhiculés entre les acteurs lors d'une modification d'un objet quelconque dans le tableau A (un message est une sorte d'explication associée à l'objet modifié).

- Garder un historique (i.e. une trace) des différentes phases¹ franchies (accomplies) en vue de pouvoir faire une évaluation et surtout de pouvoir déterminer automatiquement les objets appartenant à la phase non encore franchie. Les objets d'information possèdent différentes vues selon la phase dans laquelle ils se trouvent.
4. Les acteurs (médecins, infirmières, etc...) manipulent les objets sur le tableau A. Ces derniers peuvent *converser* entre eux en mode asynchrone et travailler directement sur les objets du tableau A de façon indépendante. Ils doivent passer, avant tout changement dans le tableau A, par des processus dits de *conversation* qui gèrent la discussion et évitent les conflits. La conversation est structurée autour du travail que les participants doivent faire.

Notons que les acteurs doivent être acceptés par des processus de protection qui vérifient les identités de ces derniers. L'accès aux objets est permis aux acteurs dont l'identité a été authentifiée et qui sont autorisés à le faire par leur statut. Pour cela, on peut imaginer une sorte de *Base de Connaissances*, spécialisée dans la gestion des acteurs, qui permet de créer des hiérarchies et de cataloguer les droits d'accès pour chaque acteur.

1. Un objet d'information passe par plusieurs phases correspondant à des transformations de son contenu.

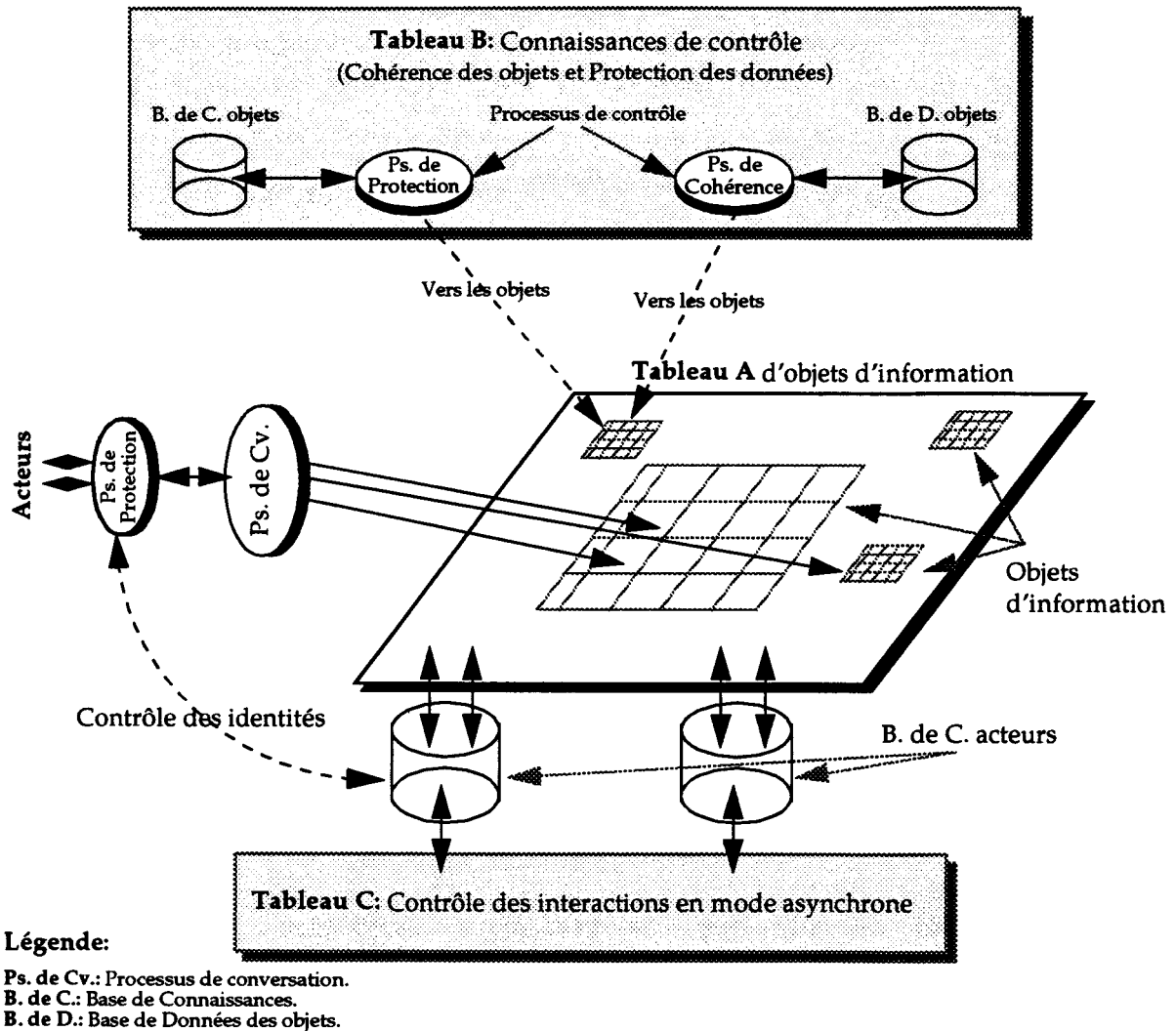


FIGURE 13 L'architecture logicielle distribuée de PLACO. Elle se compose essentiellement de trois tableaux qui sont: le tableau A contenant les objets d'information, le tableau B spécialisé dans le contrôle de cohérence des objets et le tableau C de contrôle et de gestion des interactions entre acteurs. Ces derniers, au moyen d'un processus de conversation, interagissent en modifiant et en créant des objets d'information sur le tableau A.

4.3.2 Conclusion

L'architecture que nous avons proposée pour la prise en charge des patients en USI met l'accent sur plusieurs aspects.

Elle met d'abord l'accent sur l'aspect réparti de l'information manipulée. Dans les USI, chaque patient occupe une chambre. La consultation, la prescription, ainsi que les tâches attenantes se font au pied du malade. Ceci induit la distribution de toute l'information, concernant les états de santé des patients, sur l'ensemble des chambres de l'unité médicale. Par voie de conséquence, notre architecture se doit d'être distribuée.

Les systèmes répartis ont connu une émergence sans précédent pendant les dix dernières années [Rodd91]. Ces systèmes sont classiquement utilisés comme un support de coopération¹ entre un ensemble d'ordinateurs connectés via un réseau. Le problème traditionnel dans la conception des systèmes distribués est de reconnaître l'autonomie individuelle de chaque site sur le réseau. Or, assurer une grande autonomie ne permet pas d'assurer en même temps une forte coopération. En augmentant l'autonomie d'un système, on diminue forcément le degré de coopération et vice-versa. La plupart des recherches menées dans cette optique se sont focalisées sur l'établissement d'un compromis entre ces deux extrêmes.

Dans le projet PLACO, nous n'avons pas axé notre travail sur l'aspect distribué de l'architecture. Plusieurs travaux ont déjà exploré cette problématique [Kars93] [Mull86] [Spect89] et proposé un certain nombre de solutions qui permettront de gérer l'aspect distribué de l'architecture coopérative définie dans le projet PLACO.

Le deuxième aspect de notre architecture, non moins important, concerne le contrôle. Il s'agit:

- du contrôle de cohérence (c.f. tableau B, figure 13),
- du contrôle des interactions humaines (c.f. tableau C, figure 13).

Ces deux tableaux, de concepts certes différents, font appel ensemble aux bases de connaissances et aux bases de données spécialisées pour assurer le bon fonctionnement de l'ensemble du système.

1. Le terme de coopération dénote, dans ce contexte, le degré de liaison existant entre les ordinateurs dans les systèmes distribués.

La solution pour implémenter le contrôle de cohérence peut s'effectuer à l'aide d'outils Groupware de type LinkWorks[®] [Digi93] et LotusNotes[®] [Lotu93] qui pourraient répondre à ce genre de besoin. Ils permettent de combiner aisément des bases de connaissances et des bases de données, d'assurer une approche multi-utilisateur distribuée suivant le modèle *client-serveur*, d'assurer un contrôle de cohérence des données manipulées, etc.

En revanche, le vrai problème de contrôle qui se pose dans l'architecture coopérative que nous proposons concerne les interactions humaines (prises en charge par le tableau C dans la figure 13).

D'après la classification de Rodden [Rod91], le contrôle dans les systèmes coopératifs est de deux nature: **explicite** et **implicite**.

Dans la première catégorie, on trouve:

1. les systèmes basés sur la conversation ou les actes de langage, tels que **COORDINATOR** [ATI89] et **CHAOS Project** [DeCin86],
2. les systèmes de bureautique, tels que **AMIGO** [Dani86] et **COSMOS** [Wilb88],
3. les systèmes basés sur les messages actifs semi-formels, tels que **OBJECT LENS** [Malo86b] et **ISM System** [Rodd91], etc.

Dans la seconde catégorie, on trouve:

- les systèmes de conférence médiés par ordinateur,
- les systèmes de réunion de groupes (*meeting group systems*) tels que **COLAB** [Stef87].

Le contrôle de notre système, qui est orienté vers les actes de langage, doit donc être explicite.

Par ailleurs, notre application est dédiée à être utilisée par plusieurs personnes de façon asynchrone. Elle doit donc respecter l'approche multi-utilisateur. Les acteurs doivent pouvoir se connecter de n'importe où (soit au pied du malade dans sa chambre, soit à partir d'un bureau du même service, soit d'un autre service) et n'importe quand (à n'importe quel moment de la journée).

En définitive, l'objectif du plan coopératif consiste à prendre en compte le fait que les professionnels de santé sont individuellement très performants et particulièrement conscients de l'importance de leur travail. Généralement, dans leur domaine de compétences, ils disposent de toutes les connaissances nécessaires à l'accomplissement de leurs tâches.

Mais la spécialisation croissante de chacun a comme corollaire une coordination des tâches plus délicate. Tout défaut de coordination risque de conduire à autant d'accidents ou d'incidents que nous avons réunis sous le terme de «breakdowns». L'informatisation des unités de travail devra prendre en compte cet aspect si l'on veut améliorer qualitativement la prise en charge des patients.

Notre réflexion sur l'implantation d'une telle architecture nous a amené à nous focaliser sur la modélisation du contrôle des interactions. Ce contrôle, comme le souligne RODDEN dans [Rodd93], doit être permissif et non contraignant à cause de la dynamique du travail de groupe et du groupe lui-même.

Le Modèle Workflow Asynchrone de PLACO appliqué aux USI

5.1 Introduction

Le modèle asynchrone de workflow que nous proposons se base principalement sur le *cycle workflow-action* de l'approche AWA¹ de WINOGRAD et FLORES [ATI94] tel qu'il a été présenté dans le chapitre 2. Nous avons vu que ce modèle peut s'appliquer aisément à un certain type d'organisation telle que la banque, l'administration, les compagnies d'assurance, mais qu'il s'avère inadapté pour d'autres organisations telles que les unités médicales et particulièrement les unités de soins intensifs où les acteurs ont une forte contrainte de mobilité qui impose un roulement fréquent. Un moyen de coopération efficace s'impose pour coordonner les tâches des différents acteurs dans leurs rôles respectifs.

Après l'introduction, nous consacrerons la deuxième section de ce chapitre à la présentation du modèle workflow de PLACO, en partant du cycle workflow de base de Winograd et Flores. Nous formulerons le modèle sous une forme abstraite sans tenir compte, dans un premier temps, du mécanisme de coordination. Nous examinerons l'ensemble du processus de travail, les tâches générées ainsi que les rôles qui en découlent. En revanche, nous consacrerons la section suivante à la présentation du modèle PLACO sous son aspect fonctionnel. On introduira alors la notion de *message* et nous détaillerons la

1. ActionWorkflow Approach.

façon de prendre en charge la coordination dans les USI par le simple biais des messages semi-structurés.

Définition

Dans un premier temps, nous proposons quelques définitions afin de standardiser les termes utilisés dans cette thèse et pour mieux élucider le modèle. Nous nous proposons de détailler un *workflow*, un *agent*, une *action*, un *acteur*, un *rôle*, un *processus de travail*, etc. Les définitions qui suivront s'appliquent spécialement aux termes et concepts de notre approche. Elles sont basées sur les travaux de JOOSTEN [Joos93] et de Action Technologies Inc. [ATI94].

- **WorkFlow (WF):** c'est l'unité élémentaire de travail consistant en la réalisation d'une certaine tâche. Il se compose de quatre phases: la préparation, la négociation, la réalisation et l'acceptation. Un workflow comprend deux acteurs: le demandeur de service et le fournisseur du service. On distingue deux types de workflow: principal et secondaire. Le premier workflow est créé lors du commencement d'un processus de travail. Le second est un workflow résultant. Il possède les mêmes caractéristiques qu'un workflow principal excepté qu'il est placé hiérarchiquement en dessous. On l'appelle également un *sous-workflow*.
- **Processus de travail:** c'est un ensemble d'un ou plusieurs workflow décrivant le travail dans une organisation. Par exemple, offrir un service, demander la réalisation d'une tâche, répondre à une clarification, etc.
- **Activité:** une activité est un ensemble d'événements qui se produisent sous la responsabilité d'un acteur. Une activité a est considérée réalisée si l'ensemble des événements $\{e\}$ de ladite activité sont produits.
- **Processus:** c'est un ensemble d'activités partageant un but commun. Les processus peuvent être subdivisés en sous-processus. Ainsi, une activité possède un seul acteur responsable de sa réalisation; en revanche, un processus peut nécessiter plusieurs acteurs responsables.
- **Événement:** on dit qu'un événement e déclenche une activité a lorsque la survenue de e induit la réalisation de a . Un événement résulte de la réalisation d'une activité. Par exemple, la déclaration d'un sinistre (événement) peut se faire après évaluation des

dégâts et dommages subis par une personne (activité). D'un autre côté, cette déclaration déclenche le traitement de la plainte par la compagnie d'assurance (activité).

- **Objet:** tout ce qui est ou peut être vu ou manipulé. Les objets peuvent avoir n'importe quelle forme physique et peuvent également avoir un contenu «informationnel».
- **Acteur:** on appelle acteur ou encore participant toute personne susceptible de réaliser une action faisant partie du processus de travail. Un acteur doit remplir un rôle dans l'organisation afin de s'attribuer la responsabilité d'une ou plusieurs actions.
- **Action:** on appelle action un type d'interaction qui se produit durant le cycle d'un workflow et qui change l'état de celui-ci. Une action est toujours réalisée par un acteur ou un agent. Elle représente l'élément de base d'un workflow. Une action peut être le transfert de responsabilités ou d'objets entre individus.
- **Rôle:** un acteur peut être responsable d'une ou plusieurs activités. Cette responsabilité lui attribue un rôle. Un acteur peut avoir plusieurs rôles à la fois. Un rôle peut être attribué de manière dynamique.
- **Agent:** on appelle agent une entité logicielle ou matérielle dotée d'une autonomie. Un agent possède un but et une stratégie pour l'atteindre. Il agit de façon indépendante sans aucune assistance de la part d'un acteur humain ou d'un autre agent. Il réagit aux événements qui se déclenchent dans son environnement.

5.2 Le modèle Abstrait de PLACO

5.2.1 Le modèle Asynchrone

D'après l'analyse du workflow de WINOGRAD et FLORES [Wino87b], il apparaît nécessaire avant la définition des workflow dans une organisation de:

- réorganiser l'ensemble des processus de travail,
- identifier les demandeurs et les fournisseurs de services, ainsi que les conditions de satisfaction pour chaque service,
- connaître les participants et les rôles qu'ils sont sensés remplir dans les workflow, afin de définir les responsabilités de chacun.

La question fondamentale qui se pose est de savoir si une telle approche peut être appliquée directement à des organisations telles que les USI, où la mobilité des acteurs constitue une contrainte dynamique dans le temps.

L'approche AWA est particulièrement recommandée pour les organisations dont les participants et les tâches qui leurs sont assignées sont fortement couplés. Dans ce type d'organisation, telles que les banques et les compagnies d'assurance, les tâches sont affectées de façon statique aux acteurs dans différents workflow de l'entreprise suivant leur fonction hiérarchique et leur profil. Par ailleurs, les acteurs ayant une hiérarchie bien définie dans l'organisation de l'entreprise, ne peuvent changer de façon dynamique leur fonction. Un directeur du département financier, par exemple, ne peut se substituer à un technicien ou à un opérateur et inversement. Les tâches, et par conséquent les responsabilités, ne sont pas sujettes à un transfert ou à une quelconque délégation. En réalité, ce besoin n'existe pas.

En revanche, dans les unités médicales, et particulièrement celles spécialisées dans les soins intensifs, les participants, bien qu'ils aient une fonction hiérarchique, ont souvent besoin de changer ou de déléguer leur rôle et leurs responsabilités selon les exceptions qui surgissent à un instant donné. Par exemple, un médecin accaparé par une autre tâche de priorité supérieure doit pouvoir déléguer à un autre collègue sa tâche courante. Une infirmière doit pouvoir décider d'arrêter un traitement prescrit si l'état du patient s'empire et éventuellement, lui en donner un autre en cas de force majeure. Elle a donc joué le rôle

de prescripteur bien qu'elle ne soit ni médecin ni habilitée à prescrire. Un médecin qui s'absente du service pour une raison urgente doit pouvoir déléguer ses droits et ses rôles à un collègue ou à une infirmière afin d'agir au moment opportun. De nombreuses situations en unités de soins intensifs sont marquées par cet aspect dynamique de délégation et de prévention des exceptions. Le couplage qui existe entre les acteurs et leurs tâches est donc faible et nécessite incontestablement qu'on en tienne compte dans la modélisation. Le système qui supporterait cette dynamique doit s'intéresser d'abord au rôle aux dépends de la personne physique qui le remplit.

Manifestement, il faut une extension du modèle de façon à prendre en compte cette contrainte en intégrant un mécanisme de coordination asynchrone et de gestion dynamique des rôles afin de pallier les inconvénients inhérents à la mobilité des acteurs. Pour ce faire, nous nous proposons, dans un premier temps, d'expliquer la notion de scénario coopératif dans les USI ainsi que les tenants et les aboutissants.

5.2.1.1 Scénario coopératif

Dans les unités de soins intensifs où la coopération est omniprésente, les activités humaines sont regroupées sous forme de scénarios [AIM-TANIT92] [Yous94], i.e. des structures récurrentes de travail où des ressources humaines et matérielles sont engagées. Chaque scénario possède un but, un point de départ et une fin bien définis. Chaque scénario comporte des objets tels que des données, des documents, des messages, etc. L'objectif de la description des scénarios est de faire ressortir la structure, non pas celle des objets manipulés et des tâches induites qui intéresse plus les systèmes d'information, mais plutôt celle dans laquelle le travail s'accomplit. A titre d'exemple, un scénario coopératif pourrait être «*le bilan biologique d'un patient à partir d'un prélèvement sanguin*». Ce scénario commence au niveau de la prise de décision par un médecin de prescrire une telle analyse, et prend fin lors de la disponibilité des résultats et de leur acceptation explicite par le médecin.

On se propose donc de modéliser les scénarios par le cycle workflow moyennant des extensions afin de prendre en compte les spécificités des unités médicales.

5.2.1.2 Modèle workflow étendu

Le cycle principal de workflow (c.f. figure 14) représente l'unité atomique d'un scénario. Un tel scénario garantit l'accomplissement d'une tâche homogène. Les scénarios font appel

généralement à des participants (acteurs) qui interagissent au travers de *conversations* dans lesquelles, non seulement des informations, mais en outre des engagements et des responsabilités sont partagés et transférés. Les scénarios dans les USI sont souvent formés à partir d'une combinaison de plusieurs cycles de base, un réseau de workflow. Ils décrivent les activités répétitives dans le temps de plusieurs individus.

Comme le montre la figure 14, la structure d'un cycle se compose de quatre *phases* et de quatre *transitions*. Un cycle est initié par l'acteur qui souhaite réaliser une prescription médicale (par exemple, prescrire une thérapeutique). Contrairement au modèle de WINOGRAD et FLORES, le cycle workflow adapté aux activités médicales se termine généralement avec un acteur différent de celui qui l'a initialisé à cause des «roulements» fréquents des acteurs dans les USI, ce qui complique davantage la tâche de coordination entre individus au sein d'une unité médicale. Quatre rôles d'acteurs sont impliqués dans le cycle principal: un prescripteur qui initialise le cycle par une demande de prescription, un planificateur qui planifie la prescription une fois réalisée, un exécutant ou dispensateur qui se charge de l'administration de la prescription et un évaluateur qui effectue l'appréciation clinique du résultat obtenu et conclut par un ordre vis-à-vis de l'activité en cours.

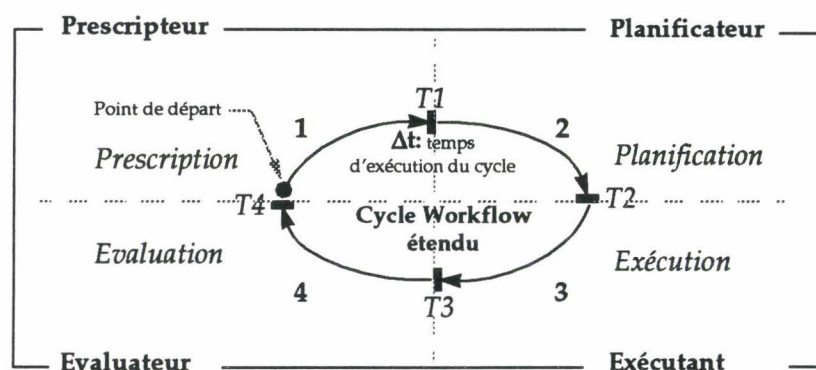


FIGURE 14 Le cycle élémentaire workflow appliqué aux unités de soins intensifs. Ce modèle identifie les actions fondamentales d'une activité de soin.

Dans ce modèle, l'accent est particulièrement mis sur les transitions qui représentent les éléments de contrôle du workflow. En effet, elles permettent le transfert de responsabilités entre les phases tout en assurant la *réalisation effective* du travail requis dans la phase précédente.

Phases du cycle WF: Elles sont au nombre de quatre et correspondent à quatre actions de base nécessaires à la réalisation d'un scénario de planification coopérative dans une unité de soins:

1. *Prescription:* lors de cette phase, un acteur (médecin en général) formule la prescription d'un traitement pour le patient.
2. *Planification:* pendant cette phase, un acteur, généralement l'infirmière-chef ou une infirmière, planifie les éléments de la prescription. Cette planification consiste en l'allocation prévisionnelle des ressources matérielles et humaines pour sa réalisation effective.
3. *Exécution:* dans cette phase, un participant à l'activité de soin, en général une infirmière, exécute la prescription. Elle administre les soins requis au patient et reporte l'impact de la prescription sur le patient. En vérité, cette phase est très particulière. Elle fait appel, dans le cas pratique, à plusieurs exécutants, car pour une même prescription, on peut avoir une dispensation répétitive dans le temps.
4. *Evaluation:* lors de cette phase, un acteur, généralement un médecin, prend connaissance du déroulement et du résultat de la prescription. Il analyse les données obtenues et conclut sur la suite à donner. Cette phase est fondamentale car elle permet de s'assurer que les différents incidents ou accidents relevés durant le cycle sont bien pris en compte par un acteur habilité à prendre des décisions.

Transitions du cycle WF: Quatre transitions représentent les éléments de contrôle du cycle. Les transitions représentent un véritable moyen de coordination des différentes tâches requises dans un cycle workflow. Elles permettent de valider le bon déroulement du cycle au moyen de transfert de responsabilités entre acteurs. Ce transfert s'effectue en rendant conscient tout acteur, impliqué dans le cycle, des actions déjà réalisées et de celles à réaliser. Une fois les conditions de transitions remplies, l'acteur de la phase suivante est averti de la tâche qui lui incombe et dont il sera responsable.

1. *A la fin de la phase de prescription,* une action est formulée et est prête à être mise en oeuvre. Le passage à l'autre phase consiste à avertir l'acteur de la deuxième phase et lui transférer la responsabilité de l'action à réaliser.
2. *A l'issue de la phase de planification,* l'adéquation entre la prescription et les moyens disponibles est vérifiée, confirmant ainsi que l'exécution effective de la prescription peut commencer. Concrètement, cela revient à dire que les médicaments et les moyens théra-

peutiques sont disponibles. L'acteur approprié est donc averti de cette nouvelle action et dispose des ressources requises.

3. *A la fin de la phase d'exécution*, la prescription est dispensée de façon effective. Un rapport sur le déroulement de cette action est disponible. L'acteur évaluateur est averti alors de la disponibilité des éléments nécessaires à l'évaluation clinique des résultats obtenus.
4. *Au terme de la phase d'évaluation*, l'ensemble du processus a été évalué et une conclusion a été formulée. Cette conclusion est une forme de décision vis-à-vis du suivi thérapeutique (arrêter, continuer ou modifier le traitement prescrit). L'acteur adéquat est averti de la nouvelle décision afin de boucler le cycle.

Remarque 4: L'administration d'une thérapeutique comporte toujours la séquence «prescription-planification-dispensation-évaluation». Ces quatre phases peuvent être réalisées par des acteurs différents, ou au contraire par un seul acteur (cas d'urgence).

Rôles, acteurs et actions induites: Dans le modèle workflow décrivant l'activité médicale dans les USI, nous avons identifié quatre rôles principaux correspondant aux quatre phases du cycle principal. Ces rôles sont remplis par trois types d'acteurs à savoir le médecin, l'infirmière-chef et l'infirmière. Par ailleurs, plusieurs actions entrent en jeu dans chacune des phases [AIM-TANIT93]:

1. Rôle de prescripteur:

- *Phase:* prescription.
- *Acteurs:* médecin ou exceptionnellement une infirmière mais sous la responsabilité d'un médecin.
- *Actions:* décision et rédaction de l'ordonnance médicale.
- *Besoins:* connaissance médicale concernant l'état du patient (disponible dans le dossier du patient).
- *Contraintes:* allergies ou intolérances possibles à certains médicaments (ces données peuvent être disponibles dans le dossier médical). Inefficacité de certains traitements. Historique médical. Disponibilité du traitement prescrit dans l'hôpital.

2. Rôle de planificateur:

- *Phase:* planification.
- *Acteurs:* infirmière-chef ou infirmière; exceptionnellement un médecin.
- *Actions:* remplir le plan de soins à partir des instructions du médecin prescripteur.
- *Besoins:* voir phase de prescription.
- *Contraintes:* disponibilité du traitement dans la pharmacie de l'hôpital.

3. Rôle d'exécutant ou de dispensateur:

- *Phase*: exécution ou dispensation.
- *Acteurs*: infirmière.
- *Actions*: dispensation du traitement médical au patient.
- *Besoins*: le plan de soins
- *Contraintes*: respect rigoureux du plan avec prise en compte des réactions du patient.

4. Rôle d'évaluateur:

- *Phase*: évaluation.
- *Acteurs*: médecin (soit le même que dans la première phase, soit un autre).
- *Actions*: prise en compte du rapport de la phase 3, évaluation clinique des effets du traitement administré au patient et prise de décision sur la suite du traitement.
- *Besoins*: le dossier patient: données historiques, paramètres fonctionnels, prescription et plan de soins, toutes les données et informations concernant le déroulement des phases précédentes (en particulier la phase 3).
- *Contraintes*: aucune.

Temps d'exécution du cycle: Une des caractéristiques importantes du cycle workflow est la contrainte temporelle qu'on peut lui associer. Il s'agit d'un laps de temps ΔT permettant, à l'issue de cette durée, de forcer la fin du cycle de workflow et par suite, d'éviter de suspendre les actions à exécuter. Ce mécanisme a l'avantage de toujours assurer la réalisation complète du cycle. Même si celle-ci est forcée, tout blocage des tâches est écarté.

5.2.1.3 Exemple de planification coopérative en USI

Le tableau 8 illustre un exemple de planification coopérative où l'on identifie quatre rôles, trois types d'acteurs et quatre actions nécessaires à l'accomplissement d'un traitement médical prescrit pour un patient. Dans la phase 1, un médecin X, jouant le rôle de prescripteur, analyse le dossier médical relatif à un patient P et lui prescrit un traitement. Dans ce cas, l'infirmière-chef ou une infirmière A analyse les contraintes de planification (dans la phase 2). Elle définit les ressources matérielles et humaines nécessaires à la dispensation du traitement prescrit. Dans la phase 3, une infirmière B, qui est désignée dans le plan de dispensation, administre le traitement au patient aux heures prescrites (réalisation effective du traitement). Cette dispensation a lieu dans la phase 3 et peut être répétitive dans le temps. A la fin de la dispensation, un médecin Y (peut être différent de celui qui a prescrit) évalue l'état de santé du malade et analyse les résultats obtenus pour

décider de la reconduction, de la suppression ou de la modification du traitement dispensé. Il doit disposer de toute l'information pertinente pour la prise de décision.

Phase	Acteur	Rôle	Action	Conclusion
Phase 1	Médecin X	Prescripteur	Analyse du dossier médical	Prescription d'un traitement
Phase 2	Infirmière-chef ou infirmière A	Planificateur	Analyse des contraintes de Planification	Plan d'action pour la dispensation
Phase 3	Infirmière B	Dispensateur	Réalisation du traitement médical	Administration du traitement au patient
Phase 4	Médecin Y	Evaluateur	Evaluation de l'état du patient	Reconduction, modification ou suppression de la prescription

TABEAU 7 *Exemple de rôles, types d'acteurs et actions requises pour la planification coopérative d'un traitement médical en USI. Une prescription d'un médicament entraîne une planification des ressources humaines et matérielles pour une dispensation effective et efficace. L'évaluation des résultats obtenus tient compte des ressources utilisées ainsi que des données recueillies lors des phases précédentes.*

En réalité, ce modèle, qui décrit les quatre phases et les différentes actions identifiées, a ses limites. Il permet de décrire les tâches et les rôles principaux nécessaires à la réalisation de la planification coopérative des traitements en USI. Dans la modélisation, nous avons supposé que le cycle principal de workflow est suffisant pour décrire les activités en USI. Ainsi, dans une planification coopérative médicale, un acteur prescrit, un deuxième planifie la prescription, un troisième dispense le traitement prescrit et un dernier évalue les résultats obtenus. Par conséquent, on présuppose que tout rôle affecté sera rempli dans les meilleurs des cas et qu'aucune contrainte ne pourra s'interposer et empêcher un acteur de réaliser sa tâche. Or, dans la réalité, le scénario de la planification médicale se déroule autrement et est certainement plus complexe que ce que décrit le cycle principal de workflow. Les acteurs ne se limitent pas à une boucle de *prescription-planification-dispensation-évaluation*. L'étude et l'analyse psycho-sociologiques, réalisées en USI [AIM-TANIT92] (voir chapitre 4), a révélé que dans la planification coopérative en USI, les acteurs sentent souvent le besoin de *déléguer une de leurs tâches* à d'autres collègues car ils ne disposent pas, par exemple, du temps nécessaire à la réalisation d'une tâche, ou qu'ils ne possèdent pas les connaissances et/ou les ressources requises pour mener à bien leur

tâche courante. D'autre part, souvent, ils sont appelés à *demander des clarifications informelles* pour obtenir des informations complémentaires afin de faciliter l'accomplissement effectif de la tâche en question. En résumé, les acteurs ont besoin d'établir des *conversations asynchrones* entre eux pour déléguer leurs rôles et clarifier leurs tâches. Ces contraintes de fonctionnement, dynamiques de surcroît, s'opposent à l'hypothèse préétablie et peuvent empêcher la réalisation de la planification médicale. Le modèle ne se limitera donc pas au cycle principal de WF et doit forcément tenir compte de ces contraintes.

D'un autre côté, WINOGRAD, dans son analyse de *conversation pour action* [Wino87a], définit, entre autres, la conversation pour la délégation et celle pour la clarification comme étant deux actes de langage à part entière. Chacun des deux actes se traduit par une conversation établie entre deux parties (un locuteur et un auditeur). Ces deux actes de nature conversationnelle, forment donc deux cycles de workflow supplémentaires qui doivent s'ajouter au cycle principal de workflow. Les phases de chaque cycle sont la préparation et l'émission de la demande, la mise en accord sur les conditions de satisfaction (qui est souvent implicite), la réalisation de la demande (réponse affirmative ou négative donnée par l'auditeur), et finalement l'acceptation ou le refus du résultat par l'émetteur de la demande. Nous appellerons ces deux cycles *Délégation et Clarification*.

De façon plus générale, il existe plusieurs et diverses formes de conversations pour action dans les activités hospitalières [AIM-TANIT92]. Certaines conversations sont de nature routinière telle que l'administration de la prescription. D'autres le sont moins, telles que la délégation et la clarification du fait de leur nature dynamique. Le besoin de déléguer ou de clarifier n'est pas présent de façon systématique pour chaque prescription d'un traitement.

D'après WINOGRAD, «un système basé sur la perspective de la conversation pour action permet d'assurer le contrôle de l'accomplissement de tous les actes de conversation prédéfinis au moyen d'un médium de communication» [Wino87b].

Nous développerons, dans ce qui suit, les deux sous-cycles workflow de délégation et de clarification pour mieux comprendre leur utilité et leur apport quant à la modélisation des activités en USI.

5.2.2 Sous-Cycle de Délégation

Concrètement, on a besoin de déléguer une tâche T à cause des contraintes imposées qui s’opposent à la réalisation de T . En fait, la délégation de T représente un sous-cycle au regard du cycle principal de WF décrit dans le paragraphe 5.2.1. La conversation qui lui est associée est sous-jacente à la réalisation de la tâche T du cycle de WF principal (voir figure 15). Elle est associée à la phase principale dans laquelle T est réalisée. Le sous-cycle de délégation est appelé aussi un cycle **secondaire**. La *conversation pour délégation* permet d’anticiper les «breakdown» relatifs au blocage ou à la suspension du processus de travail dans les organisations où les participants sont particulièrement mobiles. L’acte de délégation fournit un degré de liberté supplémentaire au système supportant la coordination des tâches. Il permet aux acteurs de se substituer aux autres dans les situations critiques préjudiciables au patient ou à l’organisation de l’unité médicale.

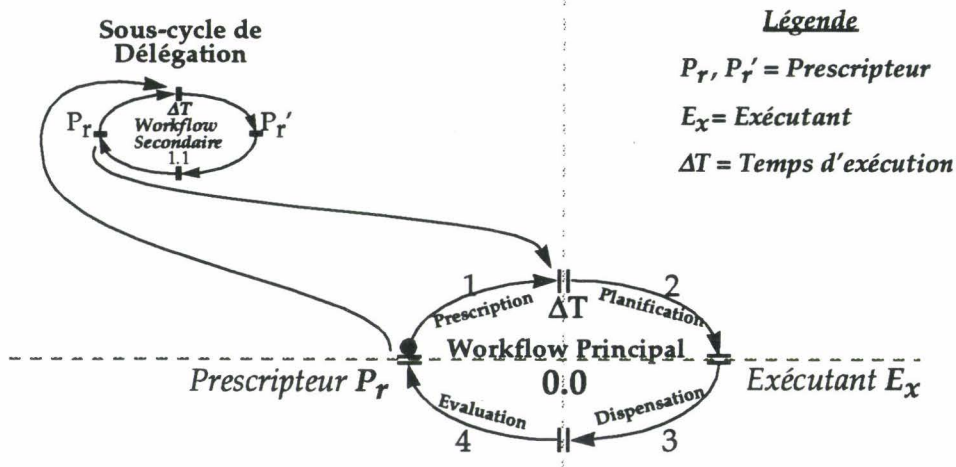


FIGURE 15 Le cycle principal de WF appliqué aux unités de soins intensifs avec le sous-cycle de délégation attaché à la phase de prescription. Le cycle 1.1 s’appelle un sous-cycle ou un cycle secondaire par rapport au cycle principal 0.0.

Dans la figure 15, le sous-cycle est initialisé par l’acteur prescripteur P_r qui demande une délégation de la prescription courante à l’intention d’un autre acteur P_r' habilité à prescrire. L’acteur demandeur doit déclarer satisfaction pour que le sous-cycle prenne fin. Autrement, le temps d’exécution du sous-cycle intervient et force la fin de la délégation. La satisfaction du demandeur peut être une acceptation ou un refus de délégation.

5.2.3 Sous-cycle de Clarification

La clarification permet aux participants de faire face aux «breakdown» concernant l'interprétation des conditions de satisfaction d'une action, ou de les anticiper. Ces conditions sont toujours interprétées selon une connaissance implicitement partagée par les acteurs [Wino87a]. Or, ce partage de connaissance est partiel et a besoin d'être négocié. WINOGRAD illustre la *conversation pour clarification* par un exemple médical très intéressant. Il énonce que la demande «donner au patient P un peu de Diazine» pourrait entraîner des questions telles que «tout de suite ou avec les médicaments du matin?» ou bien «quel dosage?». Dans les USI comme dans d'autres organisations, la précision de l'information fournie n'est jamais garantie. La précision est encline à différentes interprétations qui sont relatives à chaque acteur. Elle dépend du degré de partage du «background» requis pour l'accomplissement de l'action demandée.

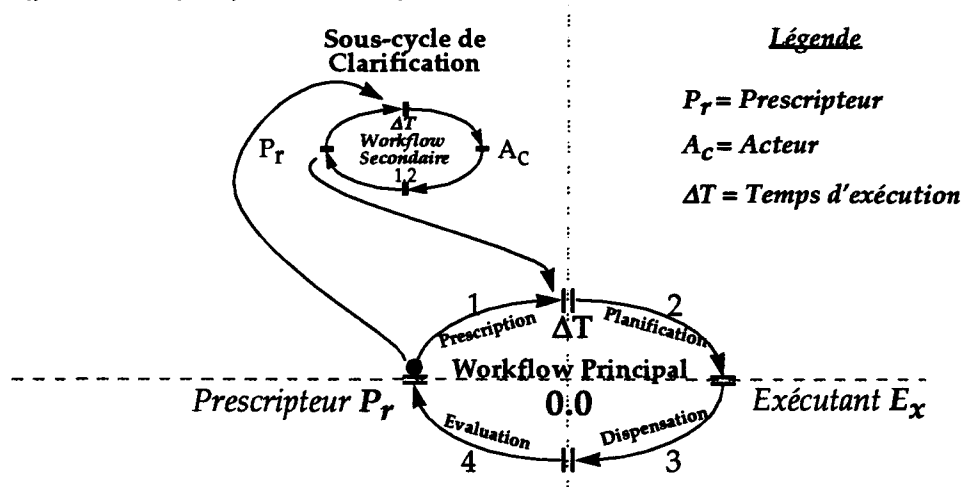


FIGURE 16 Le cycle principal de WF appliqué aux unités de soins intensifs avec le sous-cycle de clarification attaché à la phase de prescription. Le cycle 1.1 s'appelle un sous-cycle ou un cycle secondaire par rapport au cycle principal 0.0.

Un acteur demande la clarification d'une action \mathcal{A} car il ne dispose pas de l'information nécessaire à la réalisation de \mathcal{A} . Comme pour la délégation, la clarification de \mathcal{A} représente un sous-cycle au regard du cycle principal de WF. La conversation qui lui est associée est sous-jacente à la réalisation de l'action \mathcal{A} du cycle de WF principal (voir figure 16). Elle est associée à la phase principale dans laquelle \mathcal{A} est réalisée. Le sous-cycle de clarification est appelé également un cycle secondaire. Selon qu'il s'agisse d'une demande ou d'une offre, le sous-cycle est initialisé par l'acteur prescripteur P_r qui demande ou offre une

clarification sur la prescription courante à l'intention d'un acteur du cycle de workflow. Cet acteur peut être l'évaluateur E_v (phase précédente), le planificateur P_n (phase suivante), etc.

Quand il s'agit d'une demande de clarification, l'information requise concerne généralement la dernière évaluation accomplie dans la phase 4 du cycle principal. En effet, l'évaluation permet de statuer sur le prochain traitement à prescrire. Dans ce cas, le prescripteur P_r peut s'adresser à l'évaluateur E_v . Par contre, quand il s'agit d'une offre, le prescripteur P_r peut s'adresser à un autre acteur, par exemple, l'acteur qui exécute l'action suivante. Celle-ci requiert l'information relative à la prescription réalisée dans la première phase.

5.2.4 Propriétés des liens des sous-cycles de Délégation et de Clarification

Nous nous proposons donc d'étendre le modèle en attachant, dans un premier temps, les deux sous-cycles de délégation et de clarification à la première phase du cycle principal, et ensuite, en généralisant ces deux sous-cycles à l'ensemble du modèle.

Les deux cycles secondaires sont dynamiques dans le temps et ne peuvent être figés de façon statique dans le modèle car, aussi bien la demande de *délégation* que celle de *clarification*, entrent en jeu dès qu'un besoin est explicitement exprimé par l'acteur chargé de la tâche en cours. Le sous-cycle de délégation est dynamique du fait que la demande associée a lieu en cas d'impossibilité pour un acteur d'accomplir convenablement sa tâche pour une certaine raison. Le sous-cycle de clarification entre en jeu si l'information requise pour l'accomplissement d'une tâche n'est pas disponible ou est incomplète. Ce besoin, que ce soit en matière de délégation ou de clarification, peut exister à n'importe quel moment de l'exécution du cycle principal.

Comme illustré dans la figure 17, l'acteur P_r , qui a pour rôle de prescrire un nouvel objet, dispose de trois actions principales dans le cadre de son activité. Il peut exécuter, déléguer ou clarifier une tâche. On associe alors à chaque action des liens ayant des propriétés particulières. Dans ce qui suit, nous détaillerons les trois actions avec les liens correspondants.

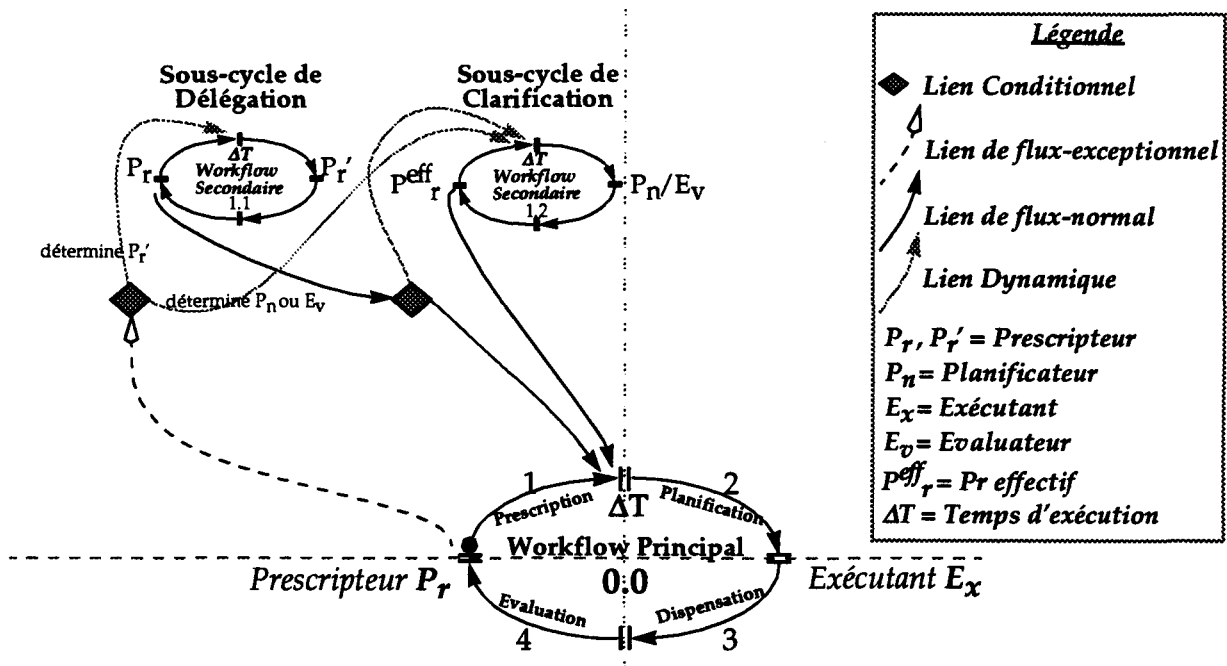


FIGURE 17 Le cycle workflow principal appliqué aux unités de soins intensifs avec les sous-cycles de délégation et de clarification. Ce modèle identifie les actions fondamentales d'une activité de soin. L'acteur est capable d'exécuter la tâche de prescription s'il dispose de toute l'information nécessaire à l'accomplissement de la tâche, de déléguer sa tâche à un autre acteur, et/ou de demander une clarification afin de disposer de l'information indispensable à la réalisation d'une prescription. Il peut également générer un arrêt de prescription (faire une action nulle) si l'information obtenue à l'issue de ces deux transactions est insuffisante.

5.2.4.1 Exécution d'une tâche

L'acteur peut *exécuter*, à proprement parler, la tâche qui lui incombe, à savoir la prescription d'un nouveau traitement. C'est la situation normale si l'acteur dispose de toute l'information nécessaire à l'accomplissement de sa tâche. Dans ce cas, le lien existant entre cette action et le cycle principal est de type *flux-normal* (représenté en trait continu).

L'acteur prescripteur peut générer un arrêt (ou un échec) lors de la prescription. Plusieurs raisons peuvent rendre une prescription impossible: modification de l'état du patient, acteur incapable d'assumer le rôle pour indisponibilité de ressources, etc. L'arrêt de prescription impose forcément, dans ce cas, une évaluation. On «court-circuitera» les deux phases de planification et d'exécution car elles sont obsolètes et insignifiantes.

Remarque 5: L'échec ne figure pas explicitement dans le modèle, car il est assimilé à un résultat plutôt qu'à une action provenant d'une transaction conversationnelle.

5.2.4.2 Délégation d'une tâche

L'acteur peut *déléguer* sa tâche de création de prescription à un autre collègue lorsqu'il ne possède pas les connaissances nécessaires ou qu'il est accaparé par une autre tâche de priorité supérieure. Ainsi, on évitera de tomber dans des situations de blocage. Dans ce cas, un sous-cycle conversationnel de type demande de *délégation* est généré *exceptionnellement* à l'intention d'un acteur P_r' habilité à remplir le rôle de prescripteur (typiquement un médecin). Ainsi, et comme l'illustre la figure 17, la délégation est liée au cycle principal par un lien de type *flux-exceptionnel* (représenté dans la figure 17 par un trait discontinu).

La recherche d'un acteur P_r' se fait de façon dynamique car les contraintes imposées à chaque participant évoluent dans le temps et dépendent fortement de la charge du travail dans l'unité de soins. On ne peut alors que faire une *attribution dynamique* des rôles aux différents acteurs potentiellement disponibles.

Le fait de décider, à un instant donné, qu'un acteur **A** est à même de remplir un rôle **R** est sujet à modification au fil du déroulement du workflow principal. On définit alors la notion de *lien dynamique entre l'acteur et le rôle* qu'il est sensé remplir. Ce lien a pour propriété de générer dynamiquement l'acteur P_r' en se basant sur les contraintes de disponibilité de chaque participant. Ce lien permet de déterminer l'acteur le mieux placé et ayant le moins de contraintes de disponibilité vis-à-vis de la tâche à accomplir. Il est représenté, comme l'illustre la figure 17, en trait gris continu.

La délégation de tâche s'avère donc déterminante et fondamentale pour notre modèle car elle permettrait d'éviter beaucoup de situations qui peuvent être préjudiciables au patient et à l'organisation globale de l'unité de soins. La recherche d'un acteur effectif P_r' se poursuit dynamiquement et le sous-cycle demeure activé tant qu'il n'y a pas rétractation ou satisfaction explicitée par l'acteur demandeur P_r . Le transfert de responsabilité se fait lorsqu'un acteur P_r' accepte explicitement cette demande et s'engage à la satisfaire.

5.2.4.3 Clarification d'une tâche

L'acteur peut demander une *clarification* concernant la tâche à réaliser en générant un sous-cycle conversationnel de type *clarification*. La transaction de clarification intervient lorsque l'acteur P_r^{eff} , prescripteur effectif, a besoin impérativement d'échanger des

informations avec un autre acteur, en l'occurrence l'acteur évaluateur E_v en vue de mener à bien la tâche de prescription. Ces informations concernent la décision prise lors de la dernière évaluation. C'est pour cette raison que la demande de clarification s'adresse à l'acteur E_v

Dans certaines situations, l'acteur prescripteur P_r^{eff} peut offrir une clarification. Dans ce cas précis, la transaction s'établit avec l'acteur P_n jouant le rôle de planificateur afin de lui fournir des informations supplémentaires indispensables à la tâche de planification. La clarification s'établit donc entre le prescripteur effectif P_r^{eff} et l'acteur de la tâche précédente E_v^{eff} ou l'acteur potentiel de la tâche à venir P_n

Comme pour la délégation, le lien associé à la clarification est de type *dynamique* et est représenté en trait gris continu (voir figure 17).

Pour passer du cycle de délégation à celui de clarification, on passe par un lien dit *conditionnel* (illustré sous forme de losange dans la figure 17). En fait, le prescripteur peut soit déléguer, soit clarifier, soit faire les deux séquentiellement. Ce choix se manifeste par l'existence d'un lien qui joue le rôle d'aiguilleur et qui dépend de la nature du besoin exprimé par l'acteur à un instant donné. A l'issue de la transaction de délégation, il a la possibilité de générer un sous-cycle de clarification ou simplement revenir au cycle principal pour effectuer le passage à la phase suivante. En revanche, à l'issue de la clarification, on s'achemine obligatoirement vers le cycle principal pour accuser la réalisation effective de la première phase.

5.2.5 Généralisation du modèle

Nous avons montré précédemment la nécessité d'avoir deux sous-cycles de *délégation* et de *clarification* pour que l'acteur P_r puisse mener à bien la tâche de prescription. Or, en réalité, ces deux transactions conversationnelles peuvent être généralisées à l'ensemble des phases du cycle principal. En effet, pour chaque phase, l'acteur en question doit disposer d'un moyen de déléguer sa tâche et/ou d'effectuer une clarification sur celle-ci. Ainsi, les trois autres phases sont similaires à la phase de prescription. On obtient alors, comme illustré dans la figure 18, trois cas supplémentaires:

1. *Dans la phase de planification*, l'acteur chargé de la planification peut avoir besoin de demander une délégation à un autre participant pour les mêmes raisons citées aupara-

vant. Il pourra également recourir à une demande ou à une offre de clarification avec d'autres acteurs afin de disposer de toute l'information requise pour l'accomplissement de la planification. La nature des liens dans la prescription reste la même pour la planification.

2. *Dans la phase d'exécution*, un exécutant dans le cycle principal peut, à son tour, se trouver dans une situation telle que la délégation et/ou la clarification pourraient lui apporter le supplément d'information indispensable à la réalisation effective et efficace de sa tâche. Par extension du raisonnement, les liens conditionnel, dynamique et exceptionnel sont les mêmes pour cette phase.
3. *Dans la phase d'évaluation*, un acteur évaluateur peut faire appel à la délégation de sa tâche et/ou à la clarification de certaines données relatives à l'accomplissement de son rôle. L'évaluation est l'une des actions primordiales car elle permet de boucler le cycle et d'aboutir à la décision concernant le prochain cycle workflow. De même, les liens reliant les sous-cycles au cycle principal restent les mêmes que pour les trois autres phases.

Ce raisonnement est valable chaque fois qu'un acteur, se trouvant dans n'importe quelle phase du cycle principal, s'engage à remplir son rôle. De cette façon, on peut penser prévenir tout écart et tout incident. Le schéma global de la figure 18 illustre les sous-cycles liés dynamiquement au cycle principal de workflow. Il montre l'ensemble des actions possibles dont dispose un acteur dans le cycle principal: *exécuter* la tâche courante liée à la phase du cycle principal si l'acteur est disponible et dispose de la connaissance requise; *déléguer* sa tâche s'il est accaparé par une tâche plus prioritaire ou s'il ne dispose pas de la connaissance nécessaire; *clarifier* sa tâche en fournissant ou en demandant le complément d'information indispensable à la réalisation de la tâche courante ou la tâche suivante.

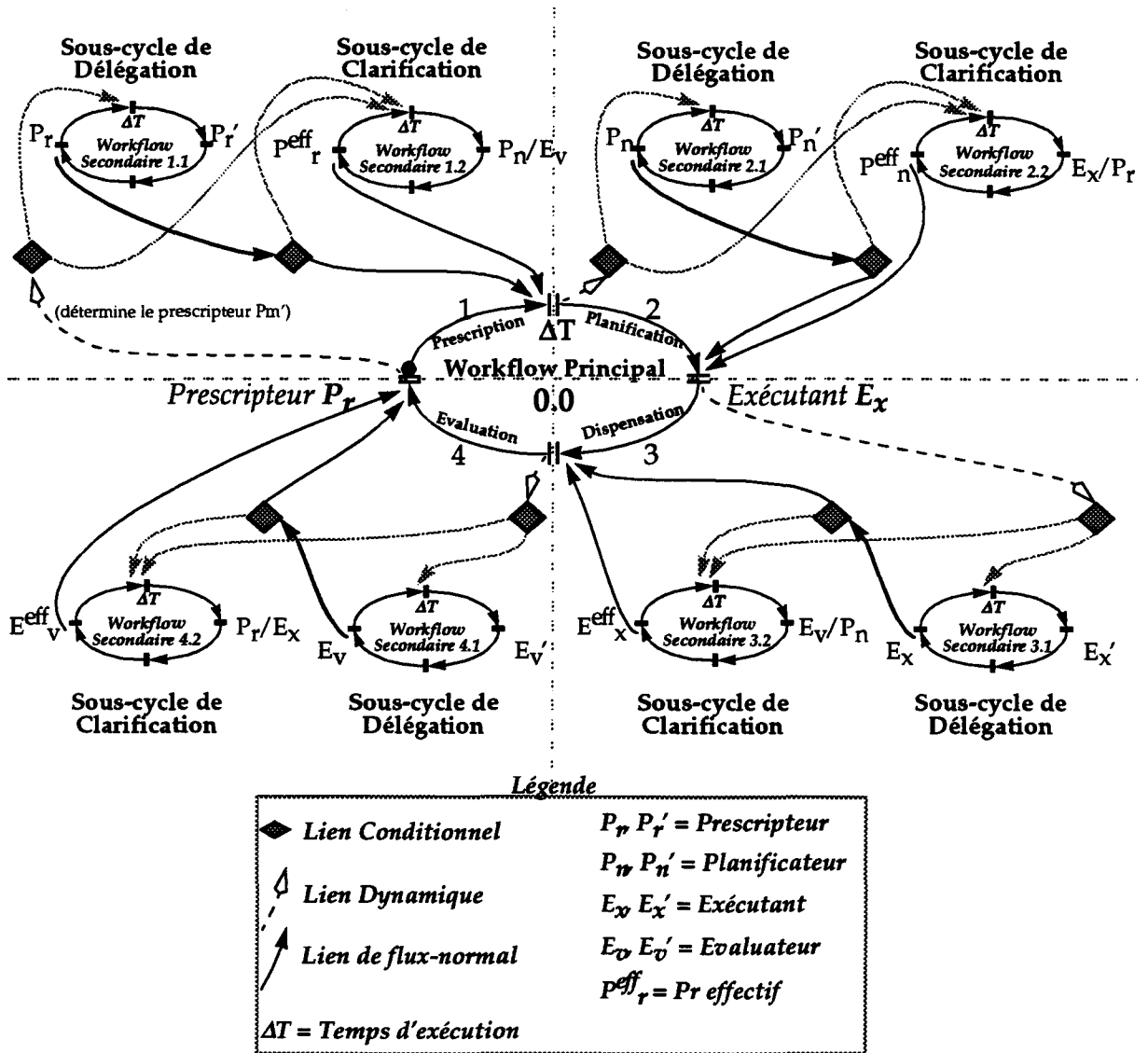


FIGURE 18

Les sous-cycles de délégation et de clarification généralisés à toutes les phases du cycle principal 0.0. Les liens dynamiques et conditionnels reliant ces sous-cycles mettent en évidence la possibilité, à tout moment, pour un acteur de déléguer sa tâche, demander ou fournir une clarification concernant son rôle. La conversation de type délégation s'établit avec un acteur habilité à remplir le même rôle que l'acteur demandeur. La clarification se passe avec l'acteur responsable de la tâche précédente dans le cycle principal, ou avec celui qui, potentiellement, sera le responsable de la tâche suivante.

5.2.6 Exemple d'un cas pratique de planification coopérative

Afin de mieux illustrer notre modèle, nous nous proposons, dans cette section, d'étudier un exemple pratique de planification coopérative qui est inspiré d'un cas réel. Pour des raisons d'uniformité et de cohérence des illustrations qui seront données dans ce qui suit, nous réutiliserons le même exemple chaque fois que le besoin se fait sentir.

5.2.6.1 Description et recueil des données cliniques:

Ces données sont extraites d'un dossier réel (voir annexe C) et peuvent être considérées comme représentatives. Pour des raisons de simplicité et de facilité de démonstration, nous avons éliminé quelques données répétitives et sans importance pour la validation de notre modèle. Ainsi, nous avons reconstruit les données tout en s'assurant du respect de leur cohérence et de leur vraisemblance.

Le recueil des données est fortement lié au fonctionnement interne des unités de soins intensifs. Un service de soins intensifs comprend des médecins, des surveillants, des infirmiers et des aides-soignantes. Ces différents personnels sont répartis en équipes afin d'assurer la continuité des soins durant les vingt-quatre heures. Les équipes sont généralement de deux types: équipes de jour et équipes de nuit. Les horaires de relève entre les équipes de médecins et celles des infirmiers sont généralement différents. Chaque roulement est l'occasion de transmissions écrite et orale d'information entre ceux qui partent et ceux qui arrivent. L'activité médicale est rythmée par deux temps d'évaluation principaux. Le «tour» du matin est suivi par la rédaction des prescriptions, celui du soir permet les transmissions aux médecins de garde. Une part primordiale des échanges d'informations s'effectue oralement et est à l'origine de nombreux «breakdown» cités dans le chapitre 4. Les événements médicaux importants sont normalement consignés dans l'observation médicale du patient, les événements infirmiers dans le dossier de démarche de soins infirmier. La prescription est réalisée sur les feuilles de prescription affichées au lit du malade. Ces feuilles servent également au recueil des paramètres de surveillance et à la notification de l'administration des principes thérapeutiques. De façon générale, il est difficile d'attribuer à priori telle action à telle personne. C'est une des sources de «breakdown».

Le service étudié comporte deux unités de cinq lits. Dans la journée, on y dénombre trois médecins, quatre infirmières et une surveillante, et la nuit, un médecin de garde et deux infirmières.

5.2.6.2 Construction des scénarios:

Les scénarios coopératifs ont été organisés en décrivant l'enchaînement des phases du cycle nécessaire à la réalisation des actions contenues dans le dossier clinique. Nous avons construit un scénario coopératif type qui est une *prescription simple*.

Prescription simple: Un patient est admis en USI à 00h. Le médecin de garde lui prescrit une perfusion de **dobutrex** en seringue auto-pulsée. Celle-ci est mise en place immédiatement par l'infirmière dispensatrice. Le lendemain après la visite médicale, la prescription est reconduite.

Les données apparaîtront sur la feuille de surveillance: dans la colonne de prescription et dans les cases de surveillance pour la dispensation (voir annexe C). Le médecin écrit sa prescription dans la colonne réservée à cet effet. L'infirmière inscrit les dispensations dans les colonnes correspondant aux horaires de réalisation.

Prescription thérapeutique/heures	Dispensation						
	24h	4h	8h	12h	20h	4h	12h
Dobutrex 250mg/4h	250mg/4h			250mg/8h			

TABLEAU 8 *Tableau des dispensations à faire pour un patient admis en USI.*

Le tableau 9 indique l'ensemble des dispensations administrées au patient. L'instanciation du modèle de workflow est illustrée par la figure 19. Les informations

relatives à la prescription évoluent au fil du cycle workflow et rendent compte des actions effectuées dans chaque phase du cycle.

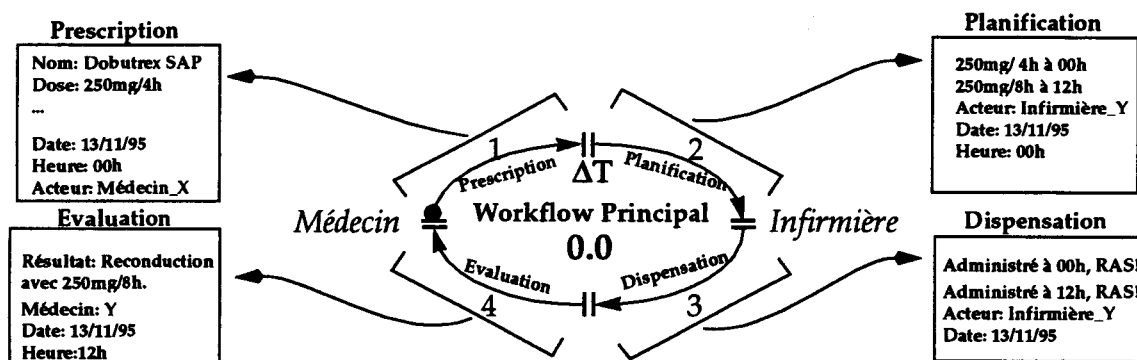


FIGURE 19 Instanciation du cycle principal de workflow pour une prescription en situation normale.

5.2.7 Conclusion

Notre modèle PLACO, du fait qu'il est basé sur le paradigme du workflow, présente une structure qui a l'avantage d'être générale et universelle. Elle est générale car elle peut être présente là où il y a coordination entre individus indépendamment de ce qu'ils font au sein d'une organisation. Elle est universelle du fait qu'elle est indépendante de toute culture, de tout langage et de tout médium de communication utilisé pour la supporter [Yous95b] [Yous95c].

Le modèle PLACO est une extension de l'approche AWA de WINOGRAD et FLORES. Il vise à supporter la coordination dans les organisations où les acteurs sont particulièrement mobiles. Dans ce modèle, l'affectation dynamique des tâches s'adressant aux rôles plutôt qu'aux individus permet [Yous95c]:

- d'améliorer notablement la coordination entre acteurs en leur offrant une vue globale des actions faites et de celles à faire,
- d'augmenter l'efficacité lors de l'accomplissement des tâches en préparant au préalable l'information utile et pertinente,
- d'accroître la productivité des acteurs en économisant le temps perdu dans la recherche de ce qu'il faut faire, et en évitant la suspension des tâches pour manque d'information ou pour indisponibilité d'un acteur,

- d'assurer le contrôle du flux d'information qui circule dans l'organisation. Ceci a pour avantage de permettre aux acteurs de se situer dans le cycle d'exécution d'une tâche donnée à un instant donné,
- de garantir un transfert correct et efficace de responsabilités en offrant les moyens de délégation et de clarification pour chaque tâche,
- de permettre aux acteurs d'être conscients sinon d'éviter les pièges de communication lors de l'interprétation du langage utilisé. Chaque acteur dispose d'une information claire, concise et précise,
- d'offrir aux participants la possibilité de générer explicitement des échecs ou des «breakdown», si la tâche en question est irréalisable, en imposant une évaluation de l'action en cours (en «court-circuitant» les phases non accomplies). Ce comportement du modèle est fondamental quant à une coordination efficace.

5.3 Le Modèle WF d'exécution de PLACO

Notre système a pour objectif de supporter la coordination de tâches entre acteurs mobiles coopérant de manière asynchrone. Il vise également à assurer le partage de l'information entre les participants. Or, les systèmes existant dans ce domaine, tels que **The Information LENSTM** [Malo86b] et **COORDINATORTM** [ATI89], utilisent comme approche fonctionnelle celle des messages semi-structurés. Les raisons en sont évidentes. L'utilisation de tels messages favorise la conception des systèmes permettant de:

- aider les gens à formuler facilement l'information qui doit être communiquée,
- sélectionner, classer et affecter automatiquement des priorités à l'information que reçoivent les individus,
- répondre automatiquement à un certain type d'information,
- automatiser ou proposer des actions que les individus souhaitent exécuter sur une certaine information reçue,
- faciliter le partage de l'information entre les personnes dans un environnement distribué,
- simplifier la mise-à-jour de l'information et de lui imposer naturellement un ordre.

Tous ces avantages font que les systèmes utilisant les messages semi-structurés se prêtent beaucoup mieux à la gestion du flux d'information et à la coordination par rapport

à ceux utilisant les bases de données distribuées. Ces derniers, bien qu'ils puissent assurer le partage de l'information, se révèlent rigides quant à la gestion automatique des priorités, de la circulation et de l'accès simple à l'information dans un environnement distribué. Notre système qui se veut distribué, faiblement couplé, et assurant la circulation et le partage de l'information doit se baser incontestablement sur l'approche des messages.

A cet effet, nous nous proposons d'utiliser ce concept simple et efficace pour supporter la coordination et aider à distribuer l'information entre participants.

Dans cette section, nous présenterons le modèle d'exécution **PLACO** avec la prise en charge par des messages issus du modèle abstrait. Auparavant, nous allons définir la notion de message que nous avons adoptée et nous détaillerons les raisons de ce choix.

5.3.1 Message semi-structuré

Nous nous baserons principalement dans notre définition de message semi-structuré sur les travaux de MALONE [Malo86a] qui a démontré que les messages semi-structurés sont «*étonnamment efficaces pour les systèmes qui supportent la coordination*¹».

Définition: On entend par message semi-structuré, un message dont le type est identifiable. Chaque type de message semi-structuré possède un ensemble de champs connus dont certains contiennent de l'information non-structurée, par exemple du texte libre.

Exemples: Un message semi-structuré peut être l'annonce d'une réunion. Dans ce cas, on peut structurer le message de façon à contenir des informations facilement analysables. On y trouve, d'après la figure 20, le destinataire, l'expéditeur, la liste CC, le sujet du message, la date d'envoi, la date de la réunion, l'heure et le lieu de la réunion. Cette information structurée pourrait servir, par exemple, à mettre à jour automatiquement l'agenda de chaque personne destinataire, en l'occurrence, les membres de l'équipe PLACO. Ce message contient aussi un champ dont le contenu n'est pas structuré (du texte libre sous forme de remarques concernant le message).

1. «*Semi-Structured Messages are Surprisingly useful for Computer-Supported Coordination*».

Message de type *Annonce_Réunion*

Nom de Champ	Valeur du champ
Destinataire	<i>Equipe_PLACO</i>
Expéditeur	<i>BEUSCART</i>
CC	<i>I</i>
Sujet	<i>Réunion_PLACO</i>
Date	<i>12/11/95</i>
Date Réunion	<i>20/11/95</i>
Heure	<i>9h30</i>
Lieu	<i>Salle_Bleue</i>
Remarques	<i>Veillez confirmer svp.</i>

FIGURE 20 Exemple de message semi-structuré. Il s'agit d'un message de type *Annonce_Réunion*, envoyé par «Beuscart» à l'intention de l'équipe PLACO pour leur annoncer la date et le lieu de la prochaine réunion PLACO. Les huit premiers champs contiennent des informations structurées pouvant être analysées facilement. Quant au dernier champ, il contient du texte libre, donc de l'information non-structurée.

On peut faire un tri automatique sur certains champs du message, par exemple sur la date de la réunion, pour pouvoir déclencher certains processus (actionner des «bips» sonores une heure avant la réunion pour un dernier rappel).

Un deuxième exemple de message semi-structuré serait l'envoi d'une clarification de prescription par un médecin «*Médecin M*» à l'intention de l'infirmière-chef «*Infirmière Y*» qui est chargée de planifier le traitement en question. Ce message doit faire référence à plusieurs données:

- le patient «*Patient_X*» pour lequel le médecin a prescrit (le message doit donc contenir un lien vers le patient «*Patient_X*»),
- la prescription «*Prescription_P*» dont il est question dans ce message (le message doit donc contenir un lien vers «*Prescription_P*»),
- la phase «*Phase_F*» à partir de laquelle cet envoi de message a été effectué (en l'occurrence, la phase de prescription d'après la structure du cycle de workflow).

On peut associer à ce message d'autres types d'information utile telle que l'heure et la date d'envoi du message, le sujet du message, etc. On aura alors la présentation schématique telle qu'elle est illustrée dans la figure 21.

Message de type *Clarification-Prescription*

Nom de Champ	Valeur du champ
Destinataire	<i>Infirmière_Y</i>
Expéditeur	<i>Médecin_M</i>
Prescription	<i>Prescription_P</i>
Sujet	<i>Clarification_Prescription</i>
Date	<i>12/11/95</i>
Heure	<i>10h30</i>
Patient	<i>Patient_X</i>
Phase	<i>Phase_F</i>
Clarifications	<i>Veillez contrôler la température du malade toutes les heures!</i>

FIGURE 21 *Exemple de message semi-structuré. Il s'agit d'un message de type **Clarification-Prescription**, envoyé par «Médecin_X» à l'intention de l'infirmière-chef «Infirmière_Y» chargée de la planification du traitement. Ce message concerne le patient «Patient_X» et la planification du traitement prescrit «Prescription_P» dans la phase «Phase_F». Le but du message est de fournir des clarifications contenues dans le champ Clarifications qui contient du texte libre (non-structuré).*

Notre choix des messages semi-structurés comme moyen de supporter la coordination de tâches entre les acteurs se justifie de la manière suivante:

- Les messages semi-structurés permettent de traiter automatiquement une grande partie de l'information. Les utilisateurs peuvent remplir un maximum d'information structurée évitant ainsi une analyse automatique du texte libre, qui peut s'avérer lourde dans l'implémentation du modèle. Par ailleurs, dans les USI comme dans bon nombre d'organisations, certaines informations sont traitées en routine. Il apparaît donc judicieux et efficace d'automatiser ce traitement afin de réduire une intervention humaine inutile. Par exemple, informer un groupe d'infirmières d'un changement dans le traitement médical d'un patient peut se faire de manière automatique.
- Les messages semi-structurés donnent la possibilité aux acteurs d'échanger entre eux des informations sans se soumettre aux contraintes de structures trop rigides. En offrant le moyen de saisir du texte libre et d'utiliser certains champs de façon non-standardisée, nous accroissons considérablement la flexibilité et l'utilité de notre modèle. En effet, dans des situations exceptionnelles en USI, une structure rigide de l'information pourrait causer un blocage du système et par suite devenir préjudiciable à la santé du

patient ou à l'organisation de l'unité. Puisque notre modèle se veut d'être un moyen original de gestion des interactions humaines, il est important donc d'offrir une certaine latitude aux acteurs.

- Les messages semi-structurés simplifient la conception de systèmes évolutifs [Malo86a]. En effet, la conception initiale et l'évolution ultérieure d'un système supportant la coopération et la communication de groupe s'avère plus facile si l'évolution se produit de façon incrémentale par des changements progressifs. Ainsi, les utilisateurs individuels pourront continuer à utiliser leur système initial sans changement s'ils le désirent. D'un autre côté, ils peuvent adopter les changements pour en tirer les bénéfices.

5.3.2 Le modèle Atomique avec les messages

Dans cette section, nous allons considérer le modèle atomique, c'est-à-dire le cycle workflow principal développé dans le paragraphe 5.2, d'un point de vue fonctionnel. Nous mettrons en place les différents messages véhiculés dans le cycle et nous expliquerons leur apport quant à la coordination des tâches entre acteurs. Nous insisterons principalement sur le rôle joué par les messages pour supporter la connaissance complémentaire indispensable à l'exécution efficace des différentes tâches du cycle WF principal.

Comme le montre la figure 22, dans chaque phase du cycle, correspondant à une tâche principale, un message M_i est généré et acheminé automatiquement vers l'acteur ou les acteurs concernés. Les messages ont pour but essentiel d'indiquer aux acteurs le ou les Rôles qui leur sont affectés. En d'autres termes, ils permettent de préciser la ou les tâches que les participants ont à accomplir à un instant donné et, si nécessaire, le complément d'information indispensable à l'exécution de ces tâches. Cependant, étant donné l'aspect dynamique des liens existant entre les acteurs et les Rôles qu'ils remplissent, ces messages ne doivent pas être *nominatifs*. En effet, les messages ne concernent pas un seul acteur. Ils s'adressent principalement aux Rôles plutôt qu'aux acteurs.

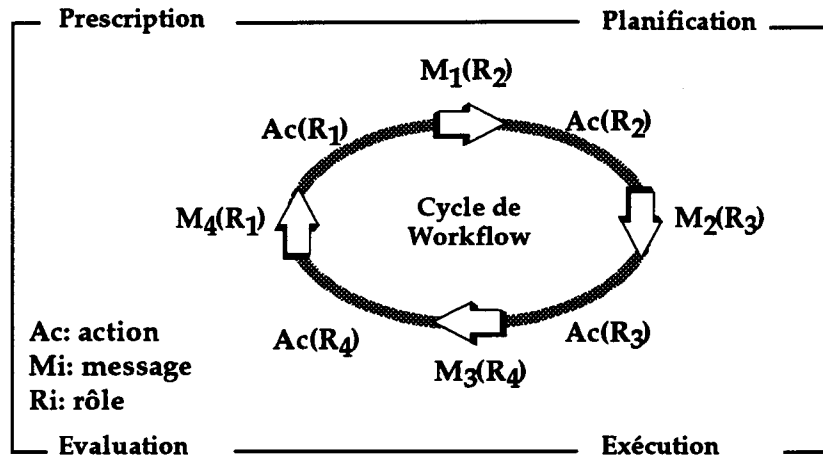


FIGURE 22

Le modèle WF fonctionnel de PLACO avec la prise en charge de la coordination par des messages. Il s'agit du cycle de WF principal étendu (adapté aux unités de soins intensifs). Dans chaque phase, un message $M_i(R_j)$ est généré par le système pour un rôle R_j ; $M_1(R_2)$ sert à spécifier dans quelles conditions l'action $Ac(R_1)$ (la prescription) a été réalisée et annonce la réalisation de l'action $Ac(R_2)$ qui est la planification. $M_2(R_3)$ annonce la fin de l'action $Ac(R_2)$ et contient l'information concernant la manière dont il faut exécuter l'action $Ac(R_3)$ qui est la dispensation. $M_3(R_4)$ indique la façon dont l'action $Ac(R_3)$ a été accomplie (i.e. la prescription a été administrée effectivement au patient) et amorce l'action $Ac(R_4)$. $M_4(R_1)$ annonce la fin du cycle de workflow. Il indique la décision obtenue quant à la suite.

Tout acteur A remplissant un rôle R peut prendre connaissance de tous les messages concernant le rôle en question. Cette approche présente l'avantage de bien différencier le rôle de l'acteur. Elle permet aux acteurs, en les informant des tâches à réaliser et des tâches déjà réalisées, d'avoir une vue globale et détaillée des différentes activités entreprises durant le cycle principal. Ainsi, un ordre de prescription d'un médicament émis par un médecin déclenchera le cycle de workflow principal. Le médecin prescripteur, informé de cette décision, procédera à l'accomplissement de la tâche qui en découle, à savoir la création effective d'une nouvelle prescription médicamenteuse pour un patient. Dans ce qui suit, nous détaillerons la nature des quatre messages générés dans le cycle workflow principal:

1. A l'issue de la première phase, le message $M_1(R_2)$ est envoyé par l'acteur R_1 ayant réalisé la tâche de prescription (P_1). Ce message est destiné à l'acteur remplissant le rôle suivant dans le cycle principal (ici, le planificateur). Ce message décrit la fin de l'action de prescription et indique l'action à réaliser en tenant compte de l'information fournie à

cet effet. En d'autres termes, il indique ce qu'il faut faire, le «quoi», dans la deuxième phase du workflow.

2. A la fin de la deuxième phase, un message $M_2 (R_2)$ est envoyé par l'acteur R_2 remplissant le rôle de planificateur (P_n). Ce message lui précise que l'action de planification de la prescription a été accomplie, et que l'action de dispensation est à faire. En termes concis, le message $M_2 (R_2)$ contient l'information concernant le «comment» de l'action de dispensation.
3. A l'issue de la troisième phase, un message $M_3 (R_4)$ est envoyé par l'acteur R_3 exécutant (E_x) à la destination de l'acteur remplissant le rôle d'évaluateur, lui précisant la manière dont l'exécution ou la dispensation de la prescription s'est effectuée réellement. Ce message précise également l'action suivante à accomplir (i.e. l'évaluation). L'information renfermée par M_3 est capitale quant à l'évaluation de la prescription.
4. A la fin de la phase 4, un message $M_4 (R_1)$ est envoyé par l'acteur R_4 jouant le rôle d'évaluateur (E_v) à la destination de l'acteur prescripteur dans la phase 1. Le message M_4 contient l'information précisant la fin du cycle de workflow ainsi que la décision concernant la continuité du traitement. L'information explicative contenue dans M_4 est indispensable à l'évaluation de la prescription. En effet, à la fin du cycle principal, l'acteur évaluateur doit disposer de toute l'information pertinente pour décider du traitement à prescrire de nouveau.

Les messages M_i , envoyés dans une phase, ont un caractère informatif et sont signés par l'acteur ayant rempli le rôle qui lui a été attribué dans la même phase. Les M_i s'adressent aux acteurs remplissant le rôle R_{i+1} (i.e. dans la phase suivante). Ce mécanisme de coordination permet de mieux préparer les actions à réaliser et de prévenir les situations où les tâches sont suspendues.

Dans la suite, on verra comment on peut étendre les messages M_i aux sous-cycles de délégation et de clarification afin de mettre en place le circuit de ces messages qui sont envoyés dans chaque phase du cycle principal ou d'un sous-cycle. Pour ce faire, on donnera d'abord le circuit des messages en prenant en compte la délégation et la clarification. Enfin, on généralisera les messages à l'ensemble du modèle abstrait schématisé dans la figure 18 (c.f. paragraphe 5.2.5).

5.3.3 Circuit des messages en situation de délégation et de clarification

Nous verrons dans ce paragraphe le circuit des messages qui tracent l'évolution du workflow dans le cycle principal et les cycles secondaires de délégation et de clarification. Nous nous limiterons, dans un premier temps, à la **phase de prescription** pour des considérations de simplification et d'allègement du schéma.

En incluant les cycles secondaires de délégation et de clarification, un flux de messages secondaires $\{m\}$ apparaît alors et s'ajoute aux messages générés dans le cycle principal (voir figure 23). Chaque message m possède un contenu:

1. **Dans le cycle de délégation**, on retrouve un circuit de messages (illustré dans la figure 23 par le flux $\{m_d\}$) qui est similaire à celui dans le cycle de workflow principal. En effet, dans la première phase du sous-cycle, un message est envoyé par l'acteur demandeur vers un autre. Ce message permet de déclencher la demande de délégation. Il s'adresse à un autre acteur habilité à jouer le rôle de prescripteur. Cet acteur est déterminé par le lien dynamique existant entre la phase de prescription et le sous-cycle de délégation (c.f. figure 17). Ce message possède toute l'information nécessaire à établir une demande de délégation de la prescription. Comme vu précédemment, cette délégation peut boucler pendant un temps ΔT tant que l'acteur demandeur n'a pas déclaré satisfaction. A l'issue du sous-cycle, un message, contenant le résultat final de la délégation, est renvoyé vers l'acteur demandeur de délégation du cycle principal. Le message représente la réponse, d'acceptation ou de refus, à la demande de délégation. Les messages générés dans le sous-cycle font partie du flux global des messages circulant dans le workflow principal.
2. **Dans le cycle de clarification**, on retrouve un circuit de messages similaire à celui du sous-cycle de délégation (illustré dans la figure 23 par le flux $\{m_d\}$). Comme le montre la figure 23, dans la première phase de ce sous-cycle, un message de type demande ou offre de clarification permet de déclencher un cycle de conversation pour la clarification. Ce message est signé par l'acteur prescripteur P_r , qui prend en charge la tâche de la phase de prescription, à destination de l'acteur planificateur P_n ou l'acteur évaluateur E_v . L'acteur à qui s'adresse ce message est déterminé dynamiquement par le lien existant entre la phase de prescription du cycle principal et le cycle secondaire de clarification (voir paragraphe 5.2). La conversation pour clarification prend fin lorsque l'acteur

demandeur déclare satisfaction du résultat obtenu ou lorsque le laps de temps ΔT est écoulé. A cet effet, un message est créé et généré à destination de l'acteur prescripteur pour lui signaler la satisfaction et lui indiquer donc la fin de la clarification. L'ensemble des messages générés dans le sous-cycle fait partie du flux global des messages du cycle principal et constitue une information complémentaire utile pour la phase de prescription.

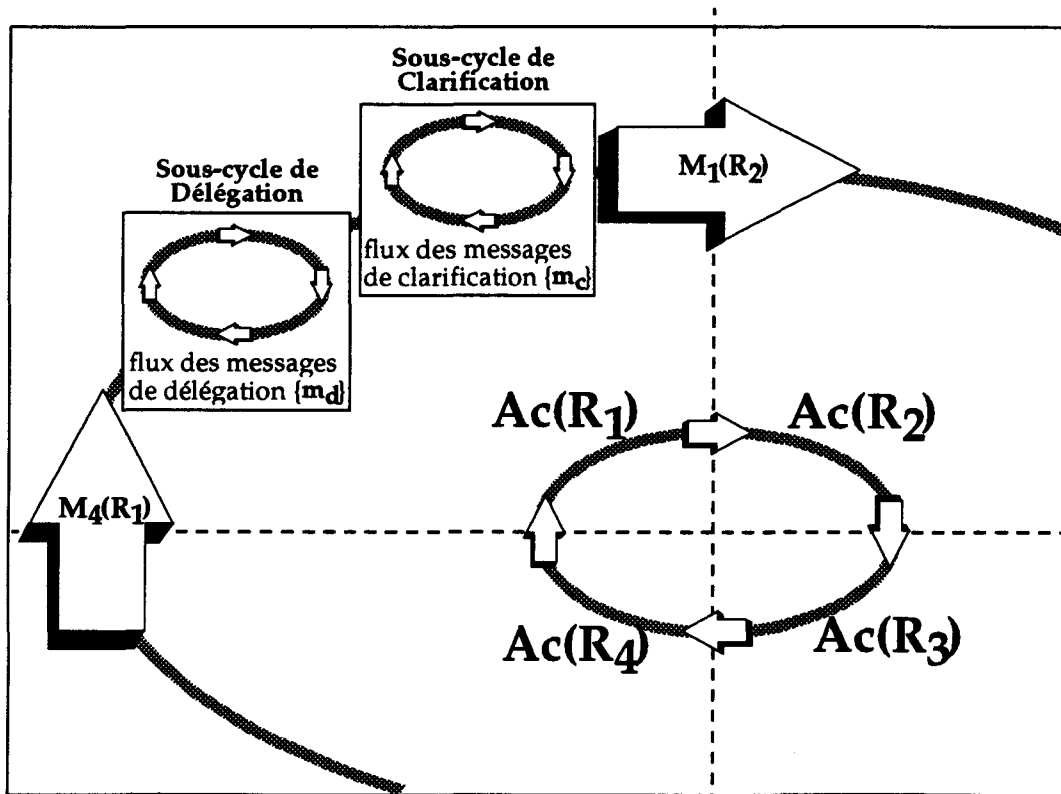


FIGURE 23 Circuit des messages générés lors de la phase de prescription dans le workflow principal. Les cycles secondaires de délégation et de clarification induisent des messages supplémentaires qui viennent s'ajouter aux messages principaux produits dans le cycle principal. L'ensemble de ces messages crée un flux permettant de tracer l'évolution de l'information.

Chaque message généré dans l'un des deux sous-cycles comprend une information dynamique et pertinente pour la suite des actions à réaliser. Ainsi, le flux de messages,

visible par les acteurs concernés, forme une couche de coordination. Elle leur permet de prendre connaissance et conscience des actions faites et des actions à faire. Cette approche basée sur les messages garantit aux différents participants une vision plus de l'information circulant dans le cycle principal adaptée à leur rôle.

5.3.4 Généralisation des messages à l'ensemble du Modèle Abstrait

Nous nous proposons de généraliser les messages à toutes les phases du cycle principal avec une prise en compte des cycles secondaires de délégation et de clarification.

Les messages $\{m_d\}$ et $\{m_c\}$ générés dans chacune des quatre phases du cycle du WF principal représentent le flux d'information dynamique qui circule le long du cycle principal. Pour chaque phase principale, un acteur est susceptible de déclencher dynamiquement des actes conversationnels de type délégation et/ou clarification (c.f. paragraphe 5.2). Ces messages sont caractéristiques de chacun des actes. Ils correspondent à l'information utile pour l'accomplissement de la phase en cours (phase à partir de laquelle les actes sont déclenchés).

Afin de mieux voir comment le flux global des messages circule dans le cycle de WF principal, nous étendrons notre explication du circuit des messages générés aux trois phases de planification, d'exécution et d'évaluation comme illustré dans la figure 24.

Dans la phase de planification, un ensemble de messages est généré dans les cycles secondaires de délégation et de clarification. Les quatre premiers messages correspondent à la *délégation de la planification*. Le premier message permet de déclencher l'acte de délégation et induit la création d'un deuxième message, ensuite d'un troisième et d'un dernier contenant la réponse à cette demande. Ces quatre messages font partie du flux global des messages circulant dans le workflow principal.

De la même façon, quatre autres messages sont générés dans le cycle secondaire de clarification. Ils renferment ainsi l'information utile et propre à la *clarification de la planification*. Comme pour la phase de prescription, les deux cycles secondaires ont une contrainte temporelle ΔT qui permet de forcer la fin de ces deux sous-cycles.

Dans la phase d'exécution, il s'agit des messages secondaires générés dans les sous-cycles de délégation et de clarification. Ils sont signés respectivement par l'acteur demandeur et

celui s'engageant à satisfaire ladite délégation. D'un autre côté, les messages concernant l'acte de clarification de l'exécution sont créés afin de supporter l'information dynamique relative à la clarification de cette tâche. La contrainte temporelle ΔT qu'on peut imposer aux deux cycles secondaires permet de forcer la fin des actes conversationnels entrepris durant ces deux cycles.

Dans la phase d'évaluation, l'analyse des messages est similaire à celles des autres phases du cycle principal explicitées auparavant.

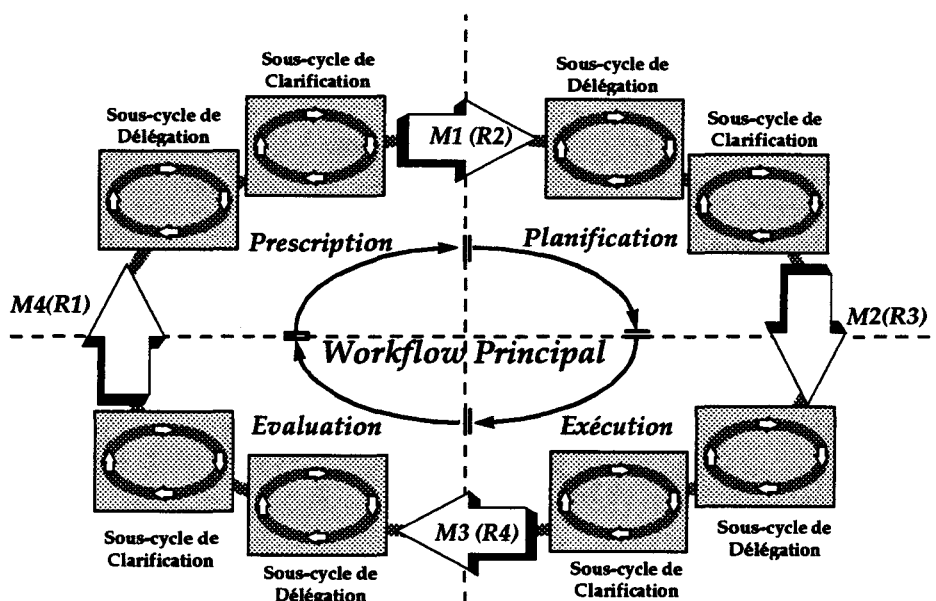


FIGURE 24 Le circuit des messages générés lors des quatre phases du cycle principal. Les messages produits dans les cycles secondaires de délégation et de clarification, pour chaque phase, créent un flux d'information indispensable pour tracer l'évolution du workflow dans le cycle principal.

Remarque 6: Les messages secondaires sont des messages dynamiques dans le temps. En effet, leur création n'a lieu que si les actes de délégation et/ou de clarification se déclenchent. Ces deux types d'actes conversationnels sont fonction de contraintes dynamiques dans le modèle global.

On dispose donc du modèle global d'exécution de workflow par messages illustré dans la figure 24.

5.3.5 Exemple d'un cas réel: instanciation du modèle d'exécution avec les messages.

Nous réutiliserons l'exemple récurrent détaillé dans le paragraphe 5.2.6. Pour chaque scénario, nous donnerons les résultats des messages générés et échangés entre les acteurs.

1. Résultats du scénario de la prescription simple:

Ci-dessous, nous présentons un tableau récapitulatif (tableau 10) contenant les messages émis lors d'une prescription d'un traitement (Dobutrex en seringue auto-pulsée) en situation normale (i.e. sans délégation et/ou clarification):

N° Message	Message_1	Message_2	Message_3	Message_3	Message_3	Message_4
Date:Heure	13:00h00	13:00h00	13:00h00	13:04h00	13:08h00	13:12h00
Phase du cycle	Prescription	Planification	Dispensation	Dispensation	Dispensation	Evaluation
Emis par	Rôle de prescripteur	Rôle de planificateur	Rôle de dispensateur	Rôle de dispensateur	Rôle de dispensateur	Rôle d'évaluateur
Reçu par	Rôle de planificateur	Rôle de dispensateur	Rôle d'évaluateur	Rôle d'évaluateur	Rôle d'évaluateur	Rôle de prescripteur
Action induite par le message	Planification	Dispensation	Dispensation	Dispensation	Evaluation	Prescription
Exemple d'information associée	Faire la planification	Faire la dispensation	RAS!	RAS!	Patient fiévreux!	Faire la prescription
Exemple de Résultat	Prescription: Dobutrex, 250mg/4h	Planification: 00h, 04h, 08h	Dispensation de 00h faite, RAS	Dispensation de 04h faite, RAS	Dispensation de 08h faite, RAS	Evaluation: Modifier la prescription

TABLEAU 9 *Tableau des messages générés pour une prescription simple en situation normale.*

Le fonctionnement du cycle entraîne la succession des messages. Chaque acteur est bien informé de la tâche qui lui incombe. Les messages générés forment un flux d'information supplémentaire et indispensable à la compréhension des différentes actions effectuées sur la prescription:

1. Le **message_1**, créé dans la phase de prescription, indique la fin de la prescription et précise la prochaine action à faire (i.e. planification).
2. Le **message_2**, envoyé à la destination du prochain dispensateur, précise que la planification est réalisé et que l'administration du traitement peut commencer.

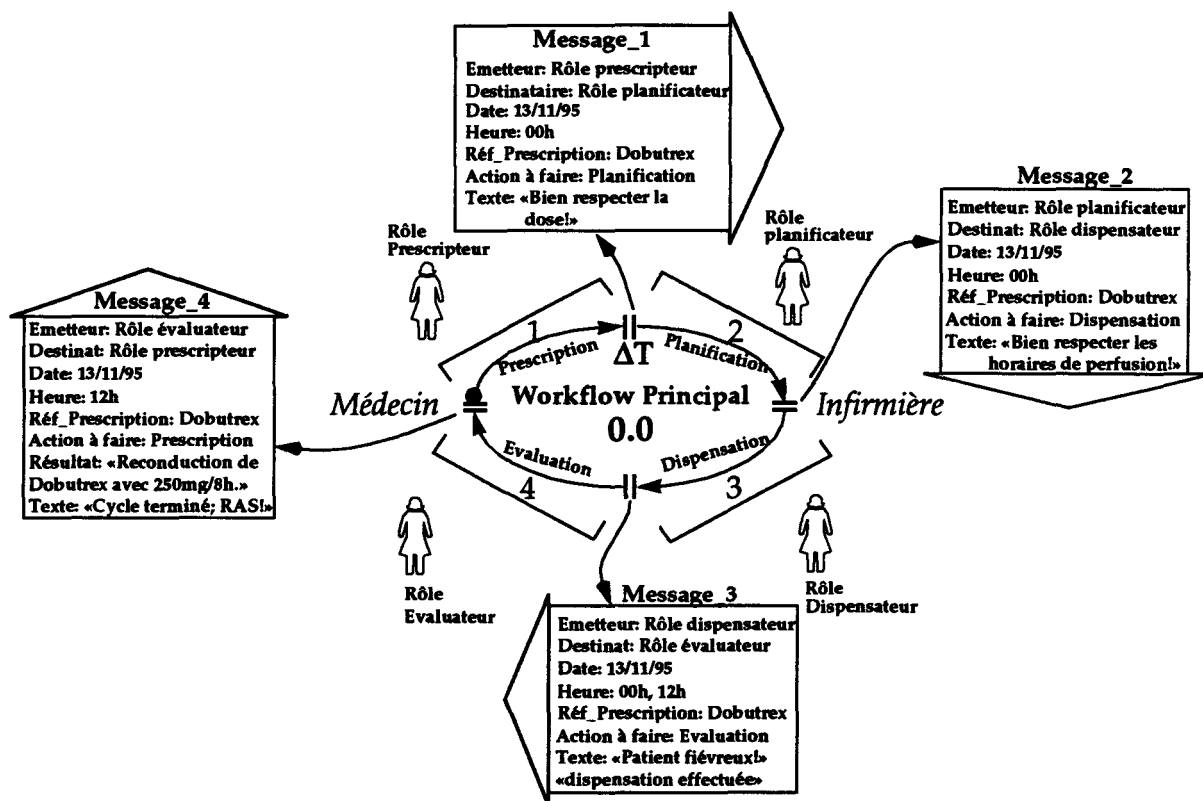


FIGURE 25 *Instanciation du cycle principal de workflow pour une prescription de Dobutrex en situation normale.*

3. Le **message_3**, envoyé à la destination de l'évaluateur, possède trois versions correspondant aux trois dispensations (car le traitement doit être administré trois fois). A chaque dispensation, un **message_3** est généré pour rendre compte de l'état de santé du patient et des conditions dans lesquelles chaque administration a été effectuée. Ces trois messages, générés dans la même phase, sont particulièrement utiles à une évaluation prématurée qui peut avoir lieu, par exemple, à la suite d'une complication de l'état du patient.
4. Le **message_4**, envoyé dans la phase d'évaluation à la destination du prochain prescripteur, contient le résultat de l'évaluation clinique du traitement administré. Une décision est alors prise pour indiquer si le traitement anciennement prescrit doit être reconduit, supprimé, modifié, etc.

La figure 25 illustre le cycle de WF principal instancié avec les différents messages générés dans chaque phase. Elle met en évidence les quatre messages qui sont générés au niveau des transitions (i.e. à l'issue de chaque phase).

5.3.6 Conclusion

Que ce soit dans le domaine de l'anesthésie ou de la réanimation, la souplesse dans la réalisation des tâches n'est rendue possible que par une forte coopération. Celle-ci ne peut se faire que si il y a un partage de l'information: transmissions orales, espaces d'informations partagées constitués par le dossier-patient, par les différentes feuilles de surveillance ou par des tableaux noirs.

Un workflow est un processus qui décrit le flux de travail en terme de tâches, d'événements et d'interactions. L'unité atomique d'un processus de travail est une boucle appelée **cycle workflow** [Wino87b] qui décrit les interactions entre deux acteurs ou agents au moyen des actions à réaliser, des rôles à jouer et des conditions à satisfaire. Chaque cycle workflow est composé de quatre phases: *préparation*, *négociation*, *réalisation* et *acceptation*. Une phase est une caractérisation de l'état du workflow basée sur l'identification des actions réalisées et celles autorisées.

Un système workflow décrit donc le réseau d'obligations et de responsabilités des personnes impliquées dans le processus de travail. Cette approche est différente de celle des systèmes d'information qui privilégie, au détriment des interactions humaines, le traitement de l'information. Cette dernière, plus classique, met en premier plan le processus d'information et, par conséquent, ne prend pas en compte la dimension humaine qui pourrait réagir de façon créative quand des exceptions surgissent inévitablement. Dans cette approche, l'utilisateur peut savoir facilement ce qui s'est passé dans les étapes précédentes. Par contre, il ne peut pas s'informer sur ce qui est requis pour les prochaines étapes. Un système workflow basé sur cette approche se révèle rigide car il ne peut s'adapter aux dynamiques des groupes et aux changements brutaux des comportements de chacun. Il se heurterait vraisemblablement à la résistance des utilisateurs. Ce point de vue est plus adapté à la conception des systèmes d'information qu'aux systèmes dédiés à intégrer la dimension humaine et à améliorer leurs interactions.

L'utilisation des messages, de façon générale, pour supporter la coordination dans les organisations où la communication est fortement asynchrone et où les interdépendances entre acteurs sont gérées et effectuées au moyen du langage (c.f. théorie des actes de langage de SEARLE [Sear69]), semble être la solution la mieux adaptée du fait de la simplicité du concept lui même et de son aspect intuitif.

Par ailleurs, l'utilisation du concept des messages semi-structurés dans le modèle exécutif, que nous avons proposé dans ce chapitre, a pu montrer dans quelle mesure nous pouvons améliorer l'efficacité et l'efficience d'un système dédié à supporter la coordination entre acteurs.

L'utilité des messages semi-structurés a fait sa preuve dans plusieurs domaines d'application; à savoir les conférences assistées par ordinateur, la gestion des calendriers, la gestion de projets, etc. Comme l'a souligné MALONE dans [Malo86a], «*les messages semi-structurés sont étonnamment efficaces pour les systèmes supportant la coordination*». Ce constat que nous retrouvons dans l'application COORDINATORTM[ATI89], basée elle-même sur la théorie des actes de langage et utilisant le concept de message semi-structuré, vient renforcer notre motivation à vouloir utiliser les messages semi-structurés. Ce concept est, de surcroît, facile à mettre en oeuvre.

6.1 Introduction

Le modèle WF proposé fonctionne au niveau de l'exécution. Il fait intervenir:

- les acteurs et les agents qui réalisent le travail en question,
- l'information présente dans leur travail,
- un mécanisme de coordination qui avertit les agents, ouvre la négociation, force les participants à respecter les accords prédéfinis et enfin, rapporte les étapes de réalisation. Le processus des responsabilités (qui fait quoi? et quand?) est, par définition, intégré dans le concept de workflow.

Nous avons choisi, pour les raisons détaillées au chapitre 5, d'utiliser comme mécanisme de coordination *les messages semi-structurés*. Ces derniers possèdent les caractéristiques suivantes:

- ils sont émis par un acteur à l'intention d'un autre,
- ils comportent des annotations qui peuvent faire référence à plusieurs types d'information,
- ils sont attachés au sujet principal qui peut être le patient, la prescription, la planification ou autre information,
- ils peuvent être partagés entre plusieurs individus sans avoir à les dupliquer ou à les diffuser en plusieurs exemplaires comme dans la messagerie classique.

Dans ce chapitre, nous traiterons l'aspect informatique: comment peut-on implanter le concept de «*tableau blanc*», déjà présenté dans le chapitre 3, comme support de partage d'information entre les acteurs de soins, en utilisant notre modèle de workflow. A cet effet, nous définirons une architecture pour l'implantation du tableau blanc et nous détaillerons les concepts utilisés dans cette définition. Ensuite, nous expliquerons dans quelle mesure le mécanisme de coordination de notre modèle, basé sur le concept de message semi-structuré, peut concourir à la structuration de l'information dynamique et à la bonne coordination des tâches en USI.

Nous finirons ce chapitre par une conclusion qui souligne les points innovants et importants de l'analyse orientée-objets de la solution PLACO.

6.2 L'informatisation du «Tableau Blanc»

Dans cette section, nous verrons quel est le type d'architecture à adopter pour l'implantation du tableau blanc, en utilisant le modèle proposé. Nous détaillerons les types d'objets qui le constituent, à savoir *objet du domaine*, *objet de contrôle* et *objet de coordination*. Par la suite, nous décrirons le «post-it» informatique tel qu'on l'entend dans PLACO: que représente un «post-it»? Quelle est sa composition? Quels types de «post-it» utilise-t-on? Quelles sont les actions réalisables sur un «post-it»? etc.

6.2.1 L'information manipulée

La plupart des approches existantes concernant la gestion du workflow (flux de travail) sont structurées autour des processus d'information. Ces approches se caractérisent par une classe d'objets d'information (formulaires, images, etc.) et définissent le workflow comme une séquence d'actions s'effectuant sur ces objets.

La structure d'organisation de base consiste en la «*circulation*» d'objets d'information entre les utilisateurs, ainsi que la spécification d'actions automatiques à considérer pendant cette opération (par exemple, dans une usine, des ouvriers peuvent être assimilés à des acteurs, et des machines à des actions automatiques). Nous ne pouvons donc pas dissocier les objets d'information des actions qui s'y exécutent.

Le modèle que nous avons défini, consistant en la gestion du workflow en USI, prend en compte l'information bien qu'elle ne figure pas explicitement dans le modèle lui-même. Elle est omniprésente mais ne représente pas le coeur de la modélisation. La modélisation de l'information est une approche différente qui conduit à la conception des systèmes d'information plutôt qu'aux systèmes de coordination.

Dans notre approche, nous définissons trois types d'information: l'Objet du domaine, le Message, et l'Agent.

6.2.1.1 L'objet du domaine

L'objet du domaine représente l'entité, à priori passive, contenant l'information utile sur laquelle vont devoir opérer les acteurs et les agents, et à laquelle, ces derniers peuvent associer des informations complémentaires.

Par ailleurs, certaines informations propres à un acteur, tels que les messages personnels, ne seraient pas pertinentes si nous les associons à un objet du domaine. Il est donc plus naturel de les lier à leur propriétaire.

Notre réflexion sur la gestion du workflow en USI nous amène à conclure que deux sortes de boîtes de messages sont nécessaires. Elles contiendront les informations complémentaires liées soit à un acteur, soit à un objet du domaine.

Ces boîtes nécessitent une gestion optimisée afin d'éviter ce qui arrive fréquemment aux personnes mobiles (abandonner leur boîtes de messages, et donc le système entier, et avoir recours au papier).

Spécification

Dans le domaine de la planification coopérative en USI, les informations partagées par les acteurs de soins sont les traitements prescrits pour les patients. Un objet-domaine de PLACO représente donc le résultat d'une prescription. Nous l'appellerons un **objet prescrit** au lieu de **prescription** afin d'éviter toute confusion. Cet objet peut être exprimé sous forme:

- de **traitement médicamenteux** (exemple, médicament **Dobutrex** en seringue auto-pulsée),

- d'**acte médical** (par exemple, une chirurgie ou simplement une surveillance de paramètres médicaux),
- d'**examen médical** (par exemple, une analyse de sang).

Un objet prescrit présente successivement cinq états correspondant:

- à un état initial,
- aux quatre phases du cycle workflow qui sont la prescription, la planification, la dispensation et l'évaluation.

D'après le graphe d'état-transition donné dans la figure 26, un objet prescrit peut être:

1. **Vide** s'il se trouve dans l'état 0. Il est sous forme d'*objet vide* d'information. Typiquement, il représente la décision de prescrire un traitement pour un patient.
2. **Prescrit** si une prescription a été faite, concernant cet objet, de façon effective. Il se trouve alors dans l'état transitoire 1.
3. **Planifié**, s'il se trouve dans l'état transitoire 2 correspondant à la phase de planification. Cet objet est prêt à être dispensé.
4. **Dispensé** s'il est a été effectivement administré au patient. Dans ce cas, il se trouve dans l'état transitoire 3.
5. **Évalué** s'il est évalué par un acteur pour disposer d'une nouvelle décision quant à la reconduite ou non de cet objet. L'état 4 est considéré final car l'objet prescrit a finit de boucler dans le cycle workflow. Il englobe alors toute l'information nécessaire et suffisante pour la suite du traitement.

La figure 26 illustre le graphe d'état-transition d'un objet prescrit dans le cycle workflow. A l'état initial, cet objet est vide car il ne contient aucune information si ce n'est le fait qu'il existe. Il passe donc au deuxième état transitoire qui consiste en la prescription effective de cet objet. Le troisième état correspond à la planification de l'objet en question. Ensuite, vient l'état de dispensation dans lequel l'objet prescrit est considéré dispensé ou administré. L'état final consiste à évaluer l'objet dispensé pour boucler le cycle de workflow et terminer par une prise de décision quant à la suite du traitement.

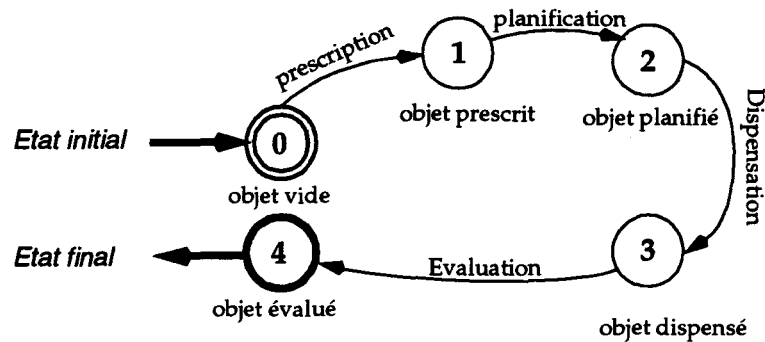


FIGURE 26 *Grappe d'état-transition d'un objet prescrit dans le cycle workflow. A l'état initial, il est vide. Les trois états qui suivent sont transitoires et correspondent aux phases de prescription, de planification et de dispensation. L'état final correspond à l'objet évalué.*

6.2.1.2 Le message

De façon générale, un message peut être défini comme un signal quelconque, simple ou élaboré, émis par un expéditeur à l'intention d'un ou plusieurs destinataires. Il possède un sens pour l'expéditeur et pour le destinataire. C'est l'élément clé de l'espace de coordination de PLACO, car il a pour rôle essentiel de conserver le complément d'information nécessaire à l'accomplissement d'une tâche par un acteur. Dans notre système, basé sur l'analyse de workflow, les messages que nous associons aux objets d'information représentent une connaissance, au sens large du terme, supplémentaire et nécessaire quant à la compréhension des changements ayant eu lieu à un instant donné. Cette approche a le mérite d'assurer une vision globale (et pas forcément grossière) du cycle entier de l'information (qui fait quoi, qui a changé quoi et pourquoi, etc.). Les messages permettent donc d'assurer une *coopération intégrante* entre acteurs. Ce type de coopération exige que chaque individu, au sein d'un groupe, puisse accéder à un espace d'information partagé qui contient une trace fidèle des tâches effectuées et leurs résultats.

Dans une messagerie classique, chaque utilisateur est identifié de façon unique et possède une boîte aux lettres qui lui est propre. Tous les messages qui lui sont envoyés sont stockés dans sa boîte aux lettres et il ne peut accéder qu'aux messages de sa propre boîte. A l'inverse, dans PLACO, outre une messagerie classique, apparaît le besoin d'un autre type de gestion des messages. Les messages peuvent dépendre du contexte et peuvent être liés à un objet du domaine. De plus, ils doivent pouvoir être adressés à un destinataire non défini (le prochain médecin ou le prochain infirmier qui s'occupera du

patient). C'est le cas d'un message signalant un retard dans l'administration d'un médicament, qui devient obsolète dès que l'administration du médicament est réalisée: il est donc lié à l'objet médicament concerné. Son destinataire n'est pas défini nommément car il s'agit du prochain infirmier qui apparaîtra dans le domaine.

Toutes ces contraintes justifient le choix que nous avons fait et qui consiste à gérer les messages comme des objets particuliers du domaine. Leur particularité tient donc dans leur capacité à être liés à d'autres objets du domaine. Lorsque l'objet auquel le message est lié est un objet représentant un utilisateur, la situation devient similaire à celle d'une messagerie classique.

Spécification

Jusqu'ici, nous avons choisi délibérément d'utiliser le terme «message» du fait qu'il est plus général. Cependant, nous verrons ultérieurement que l'utilisation du terme «post-it», à la place du message, est plus appropriée quant à notre approche.

Nous proposons donc dans la section 6.3.5 de détailler le «post-it» vu sous un aspect informatique. Nous expliquerons, par la même occasion, l'intérêt d'utiliser ce concept, plutôt que le message tout court, pour l'informatisation du tableau blanc. Nous montrerons aussi l'intérêt du «post-it» utilisé comme métaphore pour la conception des interfaces graphiques des systèmes de coordination. Nous décrirons également les différents types de «post-it» utilisés ainsi que leur moteur de fonctionnement.

6.2.1.3 L'agent

De façon générale, un agent peut être vu comme un processus qui renferme des connaissances et qui réagit dès qu'un événement reconnaissable se produit. L'agent utilise ses connaissances pour modifier et changer l'état de l'espace de travail. En partant de cette définition, on peut déduire que les agents sont autonomes, c'est-à-dire que chaque agent dispose de ses propres connaissances et ses propres règles d'activation. Par ailleurs, un agent possède un rôle, une priorité et est capable de s'activer au moment opportun. Il est capable de communiquer avec les autres agents et prendre en compte leur savoir-faire ainsi que leurs responsabilités.

Dans l'approche PLACO, Un agent est doté d'une intelligence et d'une connaissance lui attribuant ainsi un rôle et une stratégie. Nous utiliserons des agents spécialisés, par exemple, dans la gestion des messages, la gestion des objets d'information, etc.

Spécification

Le but visé par l'implémentation de la structure d'agents dans PLACO est d'automatiser l'organisation de l'ensemble des tâches effectuées sur le flux d'information. Dans la solution que nous proposons, nous distinguons un agent d'interface et deux agents-système dédiés à la gestion et le contrôle:

1. **Agent d'interface** qui est spécialisé dans la définition des fonctions de l'Interface Utilisateur Graphique (IUG) selon le rôle de l'acteur. Par exemple, une infirmière n'a pas les mêmes types d'actions et, par conséquent, les mêmes droits d'accès qu'un médecin. Par ailleurs, les acteurs ont des points de vue différents sur les objets du domaine. L'information est donc vue différemment selon le rôle de l'acteur. En outre, l'agent d'interface prend en charge les interactions de l'utilisateur avec le système via une interface graphique.
2. **Agent de workflow** qui est spécialisé dans la gestion (i.e création, destruction, mise à jour) des objets prescrits tels que les médicaments, les examens médicaux, etc. Comme vu précédemment, un objet prescrit présente cinq état différents correspondant à un objet *vide*, *prescrit*, *planifié*, *dispensé* et *évalué*. Les transitions de l'objet prescrit d'un état à l'autre sont gérés par l'agent de workflow. Celui-ci est activé par un acteur via son interface graphique. Selon l'état de workflow de l'objet manipulé, l'agent de workflow déclenche l'action requise pour amener l'objet à l'état suivant (c.f. figure 27). Par exemple, un objet nouvellement prescrit nécessite une planification de la part d'un acteur planificateur. Ce dernier active le gestionnaire de workflow, en passant par l'agent d'interface, qui prend en charge la création de l'information. Cette dernière consiste en la planification de l'objet prescrit. De la même manière, un objet planifié nécessite une dispensation. Dans ce cas, l'acteur dispensateur active l'agent de workflow qui, à son tour, déclenche l'action de dispensation. Celle-ci consiste en l'administration effective de la prescription au patient, et ainsi de suite. Les informations supplémentaires qui en découlent font changer l'état de l'objet prescrit.
3. **Agent de «post-it»**, ou encore gestionnaire des «post-it», spécialisé dans la création automatique des «post-it» et leur mise à jour. Chaque fois qu'un nouvel événement sur-

git dans l'espace d'information, l'agent de «post-it» est mis au courant par une notification de la part de l'agent de workflow. Le premier agent agit alors en conséquence et met à jour les «post-it» existants et, s'il y a lieu, procède à la création de nouveaux «post-it» pour rendre compte de l'événement en question et le signaler aux acteurs devant manipuler cette information. Nous verrons en détail les fonctions de cet agent ainsi que le fonctionnement des «post-it» dans le paragraphe 6.3.5.

Dans la figure 27, nous illustrons le fonctionnement des agents et les communications qui s'établissent entre eux. Un acteur X interagit avec son interface-utilisateur. L'agent d'interface entre en fonction et assure la transmission, sous forme de requêtes, de ces interactions à l'agent de workflow. Ce dernier, selon l'état de l'objet manipulé qui se trouve dans l'espace d'information partagé, exécute l'action appropriée sur cet objet. Par notification de l'événement qui vient de se produire (création, modification, suppression, etc.), il annonce à l'agent de «post-it» le nouveau changement survenu dans l'espace partagé. Ce dernier procède à la création de nouveaux «post-it» et à la mise à jour des «post-it» existants liés à l'objet du domaine en question et à l'acteur responsable.

Prenons l'exemple d'un acteur X qui demande de créer une nouvelle prescription D1. Il interagit, via son interface-utilisateur, avec l'agent d'interface qui transmet sa requête à l'agent de workflow. Ce dernier procède à la création effective de l'objet D1 dans l'espace d'information partagé. Quant à l'agent de «post-it», il est mis au courant de cet événement par notification de la part de l'agent de workflow. Par conséquent, il crée et met à jour les «post-it» appropriés qu'il attache à l'objet D1 pour signaler l'état de ce dernier dans le workflow, et aux acteurs destinataires pour leur indiquer le ou les actions futures à exécuter sur D1.

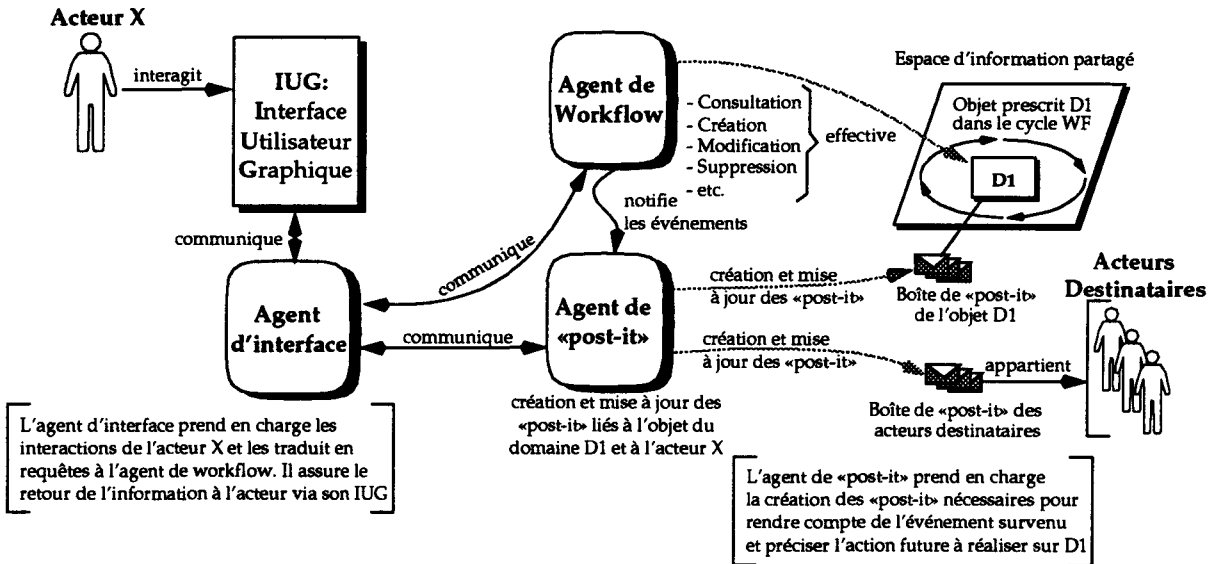


FIGURE 27 Fonctionnement des agents de la solution PLACO. L'acteur X interagit avec son IUG. Celle-ci communique avec l'agent d'interface qui prend en charge les interactions de X et les transmet sous forme de requêtes à l'agent de workflow. Ce dernier, par notification, communique à l'agent de «post-it» la nature des événements survenus dans l'espace d'information partagé afin qu'il puisse mettre à jour les «post-it» existants et, éventuellement, en créer d'autres qu'il attachera aux acteurs destinataires et à l'objet D1.

6.2.2 L'architecture proposée

L'architecture que nous proposons, et que nous appellerons PLACO, se prête bien à une implantation en couches. Cette approche n'est certainement pas nouvelle dans les systèmes de gestion du workflow, mais elle présente l'intérêt d'être facile à implanter puisqu'elle permet de faire une séparation nette entre les différents concepts utilisés. Nous proposons donc, dans la figure 28, les couches qu'englobe l'architecture PLACO et qui déterminent son fonctionnement.

A chaque couche, on fait correspondre un modèle d'objet. A la base de cette figure, nous trouvons l'objet-domaine qui représente l'espace de production ou l'espace d'information. Au dessus, on trouve le concept d'agent organisateur ou gestionnaire auquel on attribue la gestion des tâches et des événements intérieurs. Vient ensuite le message qui constitue l'espace de coordination des différentes tâches du système. Au sommet de l'architecture,

on trouve une couche jouant le rôle d'interface entre le système entier et le monde extérieur. Il s'agit de la couche des événements qui constitue l'espace d'ordonnancement.

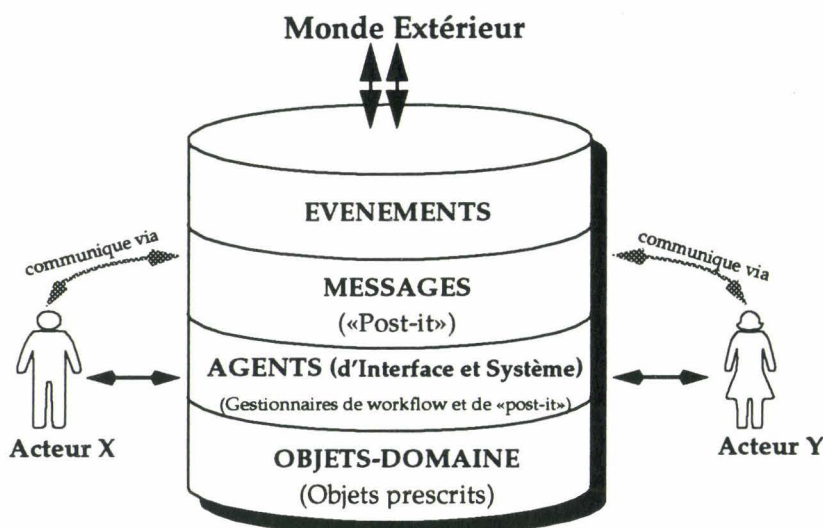


FIGURE 28 L'architecture que nous proposons est implantée en couches. Les agents d'interface sont représentés dans le système par leur interface spécifique.

6.2.3 Le «Post-it»: version informatique

La solution de «post-it» que nous avons proposée pour pallier les problèmes de coordination et de communication entre acteurs, semble avoir ses limites à cause de la gestion lourde et fastidieuse de ces derniers. Cependant, en appliquant le modèle PLACO pour l'informatisation du tableau blanc, on se rend vite compte de l'intérêt que peut apporter cette solution à base des «post-it».

Le modèle PLACO utilise les messages comme mécanisme de coordination des tâches entre acteurs. Dans les USI, les acteurs de soins utilisent les «post-it» pour, justement, renforcer la coordination de leurs activités et assurer un moyen de communication fiable. Le problème qui se pose concernant la gestion des «post-it» peut être résolu simplement et efficacement en faisant une *identification* entre le mécanisme de coordination de notre modèle qui est basé sur les messages et celui préconisé dans les USI, à savoir le «post-it». Par ailleurs, le «post-it» pourra également être utilisé comme métaphore dans la conception des interfaces graphiques pour notre solution informatique.

6.2.3.1 Définition et structure

La notion de «post-it» informatique dénote un message semi-structuré qui possède, au moins, les caractéristiques suivantes:

1. Un émetteur pourra être un acteur (e.g. un médecin, une infirmière, etc.) ou un agent-système (e.g. l'agent de workflow).
2. Un destinataire pourra être un acteur, un ensemble d'acteurs, un rôle rempli par un ou plusieurs acteurs, ou encore un objet du domaine. En effet, un «post-it», par définition, peut être attaché à n'importe quel objet au sens large du terme. Une prescription, par exemple, peut contenir un «post-it» créé par un médecin pour signaler qu'elle a été modifiée à cause d'une complication de l'état de santé du patient.
3. L'heure et la date d'émission du «post-it». L'ordre d'arrivée des «post-it» étant important quant à la gestion globale et l'ordonnancement de l'information.
4. Le type du «post-it» en question. Par exemple, certains «post-it» ont des significations différentes selon qu'ils sont créés dans la phase de prescription, de planification, de dispensation ou d'évaluation. Le typage des «post-it» facilite leur gestion et augmente leur intérêt. On peut assimiler ce typage informatique à l'utilisation, dans la vie réelle, des «post-it» de couleurs différentes.
5. Un sujet indique en quoi consiste le «post-it», et à quel contexte (i.e. phase du cycle) il se rattache.
6. Un ou plusieurs liens actifs (appelés aussi références actives) permettent aux acteurs de d'accéder directement depuis le «post-it» à l'information pertinente à laquelle ce dernier fait référence. Ceci permet d'éviter les parcours linéaires lors de la recherche des informations pertinentes.

Comme illustré par la figure 29, un exemple de «post-it» envoyé par le médecin «médecin_X» à destination du groupe d'infirmières «groupe_I1». Ce «post-it» a pour but d'annoncer aux infirmières la fin de la réalisation de la prescription «prescription_P» et les informe de l'action suivante à réaliser qui concerne la planification de cette prescription.

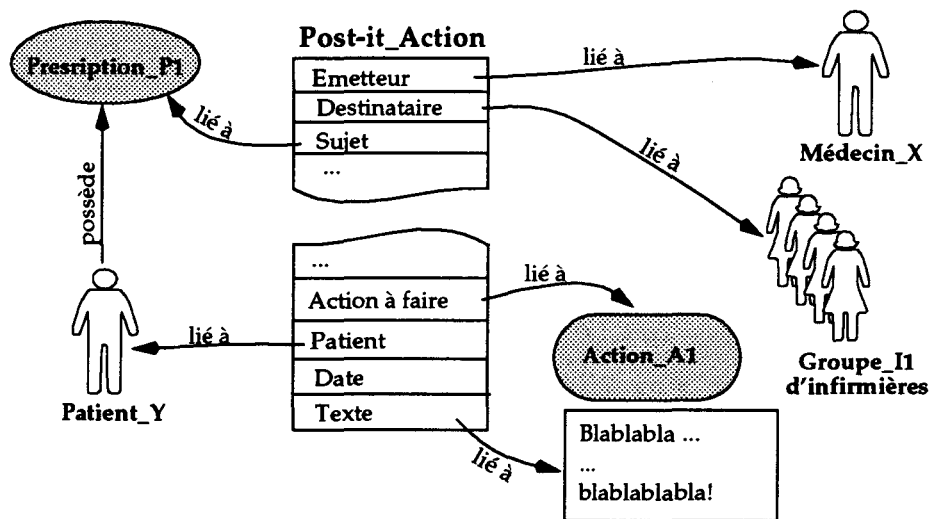


FIGURE 29 Un exemple de «post-it» créé dans la phase de **prescription**. Il contient des informations destinées à un groupe d'infirmières «groupe_I1». Il fait également référence à plusieurs objets tels que la prescription «Prescription_P1» et l'action «action_A1». Les liens de type «lié à» sont actifs et permettent aux destinataires un accès direct aux objets dits référencés.

Un «post-it» possède les propriétés suivantes:

1. Il a une durée de vie finie ou infinie. Elle peut être exprimée explicitement ou implicitement. La durée de vie peut dépendre d'un événement particulier et, dans ce cas, tant que cet événement ne s'est pas produit, le «post-it» persiste. Par exemple, un «post-it» qui annonce à un acteur la réalisation d'une action A_i reste présent tant que cette action n'est pas accomplie. Une fois A_i réalisée, un autre «post-it» vient remplacer l'ancien pour annoncer l'action suivante A_{i+1} . Un «post-it» créé lors d'une demande de délégation par un médecin doit avoir une durée de vie limitée. En effet, si l'acteur destinataire ne répond pas pendant un laps de temps ΔT , le «post-it» ne sert plus à rien et est considéré obsolète. L'acteur demandeur doit reprendre la main pour éventuellement adresser la demande à un autre collègue qui serait mieux disposé à satisfaire la demande de délégation.
2. Un «post-it» peut être attaché à un acteur, par exemple un médecin, à un groupe d'acteurs, par exemple des infirmières, ou à un objet du domaine, par exemple une

prescription. Dans la vie réelle, les individus peuvent communiquer en asynchrone et s'échanger des «post-it». Ils utilisent un support partagé pour y coller leurs «post-it» à côté de l'information référencée, ou simplement un lieu, un bureau par exemple, que le destinataire occupera vraisemblablement pendant sa présence. Ceci garantira, dans une certaine mesure, que le destinataire prendra connaissance de l'information contenue dans les «post-it».

6.2.3.2 Les «post-it» utilisés dans PLACO

Dans notre approche, nous distinguons trois grandes catégories de «post-it»:

«Post-it» de type message workflow: appelé aussi «post-it» d'information, il est créé automatiquement à partir d'une phase du cycle et est signé par l'acteur remplissant le rôle associé à ladite phase. Sa fonction essentielle consiste à transmettre l'information complémentaire nécessaire aux acteurs pour réaliser leur tâche le plus efficacement et le plus sûrement possible. Ce «post-it» contient un complément d'information qui assure la compréhension des différents événements survenus durant le cycle de travail.

Ce type de «post-it» est attaché à un objet du domaine pour rendre compte de l'état de cet objet dans le workflow (i.e. les transformations qu'il a subit) et préciser l'action future à exécuter sur ce dernier. Par exemple, lors de la création d'une nouvelle prescription pour un patient, un «post-it» workflow est créé et est attaché à cette nouvelle prescription pour préciser son état dans le workflow (i.e. phase de prescription) et l'action suivante à exécuter (i.e. la planification). Ainsi, tout objet ayant subit une transformation quelconque dans le cycle workflow (e.g. planification effectuée pour une prescription donnée) se voit attacher un «post-it» workflow.

«Post-it» de type message de coordination: il est créé automatiquement à partir d'une phase du cycle à l'intention d'un ou plusieurs acteurs susceptibles de remplir le rôle suivant dans le cycle workflow. Dans ce cas, ce «post-it» doit référencer, par des liens actifs, le ou les destinataires, ainsi que toute l'information pertinente indispensable à la compréhension de la modification survenue sur un objet d'information.

L'information contenue dans le «post-it» de coordination peut faire l'objet:

- d'une indication d'une action A et de la phase dans laquelle se trouve l'information à laquelle est lié le «post-it» (par exemple, action de planification d'une prescription P

dans la phase de planification). Le «post-it» en question peut contenir du texte libre. Par exemple: «*Svp, faire la planification pour cette prescription le plus vite possible!*»,

- d'une demande de transaction conversationnelle en vue d'une clarification ou de délégation d'une tâche T sur un objet d'information quelconque,
- d'une explication simple à propos d'une action, sur un objet prescrit, effectuée dans l'une des quatre phases du workflow. Cette information peut se révéler capitale pour le suivi du patient par les autres membres de l'équipe thérapeutique.

Cette technique, de surcroît facile à mettre en oeuvre, présente l'avantage de spécifier en permanence l'état dans lequel se trouve n'importe quel objet du domaine. Elle définit, pour un acteur, la future action à exécuter sur l'objet en question. Ainsi, les acteurs sont informés continuellement de tous les changements et modifications survenus sur un objet du domaine. De ce fait, on peut penser que l'information contenue dans un «post-it» de coordination favorise la coopération, diminue les redondances et évite l'occurrence des «breakdown» et erreurs de fonctionnement.

Comme illustré dans la figure 30, un exemple d'envoi de «post-it» de coordination adressé à un groupe d'acteurs G1 (infirmières). Ce «post-it» référence des informations tels que l'objet prescrit D1, le patient P, etc. Ce «post-it» informe le groupe G1 sur l'état de workflow de la prescription D1.

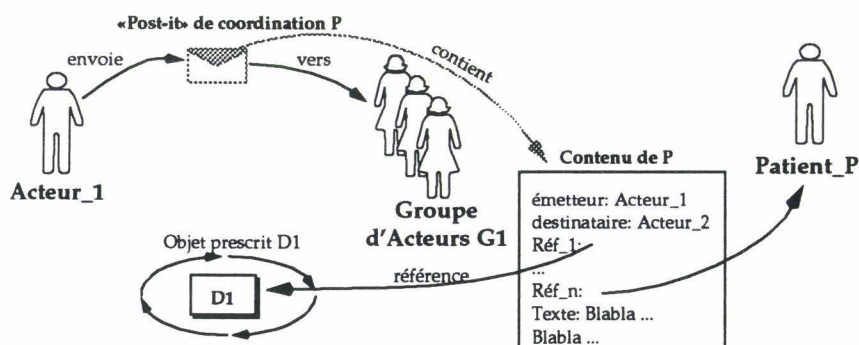


FIGURE 30 Un exemple d'envoi de «post-it» de type message de coordination attaché à un groupe d'acteurs. Le «post-it» P est envoyé par un acteur_1 vers le groupe G1 et référence plusieurs objets tels que la prescription D1, le patient P, etc.

«Post-it» de type message libre: la faculté du système PLACO de créer et gérer les messages informels, comme dans une messagerie électronique classique de type «Email», permet d'assurer entre agents une communication transversale libre. Cette propriété assure aux acteurs de s'échanger des messages libres et informels sans imposer des contraintes particulières de gestion.

Comme illustré dans la figure 31, un «post-it» P de type message informel, est émis par un acteur vers un autre et ne référence aucune information dans le cycle. Il contient du texte informel (non structuré) ne nécessitant aucun traitement automatique. Il est signé par l'acteur_1 et est destiné à l'acteur_2 à la manière d'une messagerie électronique classique.

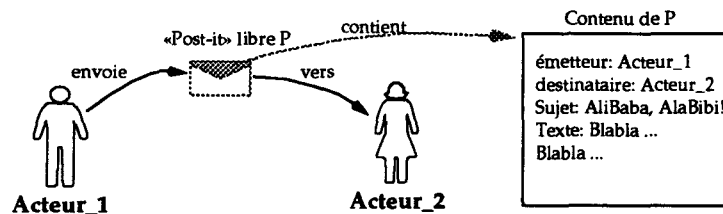


FIGURE 31 Un exemple d'envoi de «post-it» de type message libre. Le «post-it» P, envoyé par un acteur_1 vers un acteur_2, contient des informations non structurées.

6.2.3.3 Le moteur de fonctionnement des «post-it»

Comme vu précédemment, les «post-it» sont de trois types. Les «post-it» workflow, les «post-it» de coordination et les «post-it» libres:

1. Les deux premiers types sont générés automatiquement par l'agent de «post-it» suivant les événements survenus dans l'espace d'information partagé. Comme vu précédemment, ces événements lui sont communiqués par l'agent de workflow. La création de ces deux types de «post-it» a lieu chaque fois qu'une action a été ou doit être réalisée sur un objet du domaine:

- Le «post-it» workflow, attaché à l'objet du domaine, rend compte de l'état de workflow de l'objet auquel il est attaché et précise l'action suivante à exécuter sur ce dernier. Par exemple, quand un nouvel objet D de type médicament est créé, l'agent de «post-it» lui attache un «post-it» workflow pour signaler son état *vide* à

l'instant de création (voir le paragraphe 6.3.1), ainsi que pour préciser l'action suivante à effectuer (i.e. la prescription) sur l'objet D.

- Le «*post-it*» de *coordination*, attaché aux acteurs, possède une référence à un objet du domaine. Il sert à spécifier la tâche suivante T à réaliser sur l'objet en question. Ce «*post-it*» peut être destiné à un ou plusieurs acteurs qui, potentiellement, peuvent remplir la tâche T.

Les deux «*post-it*» sont générés simultanément. Ils ont pour objectif d'assurer une meilleure coordination des tâches entre acteurs. Ainsi, un acteur peut prendre connaissance de l'état d'un objet du domaine de deux façons:

- soit par le biais du «*post-it*» attaché à l'objet lui-même. Dans ce cas, l'acteur doit accéder d'abord à l'objet du domaine pour pouvoir consulter le «*post-it*» en question. Comme précisé auparavant, ce «*post-it*» renseigne les acteurs sur l'état de workflow de l'objet.
 - soit par le biais du «*post-it*» qui est attaché à l'acteur lui-même. Dans ce cas, l'acteur n'a pas besoin d'accéder d'abord à l'objet en question. Il est directement renseigné de l'état de l'objet et de l'action à exécuter sur ce dernier par l'intermédiaire du «*post-it*» de coordination qui lui est attaché.
2. Le troisième type concerne les «*post-it*» libres qui sont générés «manuellement» par les acteurs telle que dans les messageries électroniques classiques. Leur gestion et leur fonctionnement est donc similaire à ceux du «**Email**». Un «*post-it*» libre ne peut être attaché qu'aux acteurs. Leur création est explicite et n'est sujette à aucune gestion automatique. L'agent de «*post-it*» n'entre en jeu que sur demande explicite de la part de l'acteur.

6.2.4 Exemple de scénarios avec gestion de «*post-it*»:

Nous proposons, ci-dessous, l'exemple récurrent proposé dans le chapitre 5. Il s'agit des scénarios de prescription, de planification, de dispensation et d'évaluation du traitement **Dobutrex** en seringue auto-pulsée pour un patient P en situation normale.

1. **Prescription:** la figure 32 présente un scénario de prescription d'un nouvel objet (prescription du traitement **D1: Dobutrex** en Seringue Auto-Pulsée). Averti par ce nouveau changement, l'agent de prescription crée l'objet prescrit **D1** dans l'espace d'information partagé et informe l'agent de «*post-it*» de ce changement. Ce dernier crée et attache un

«post-it» workflow P1 sur le nouvel objet D1 pour indiquer son état courant dans le workflow (i.e. phase de prescription). Il crée également un deuxième «post-it» R1 de coordination qu'il «attache» aux acteurs destinataires susceptibles de remplir le rôle de planificateur et leur annonce l'action future à réaliser sur le nouvel objet (i.e. action de planification). Ce «post-it» fait référence au premier «post-it» P1 et donc au nouvel objet prescrit D1.

L'acteur jouant le rôle de planificateur possède deux moyens lui permettant de prendre connaissance de l'action future à réaliser. Le premier moyen est direct car il se fait via le «post-it» R1 qui lui est proprement attaché et qui référence l'objet en question. Le deuxième moyen est indirect puisqu'il se fait grâce au «post-it» P1 appartenant à l'objet sur lequel l'action doit s'exécuter. Dans le deuxième cas, l'acteur doit d'abord accéder à l'objet avant de connaître l'action à exécuter. Tandis que dans le premier cas, l'acteur est informé directement de l'action à effectuer sans avoir à accéder à l'objet en premier lieu.

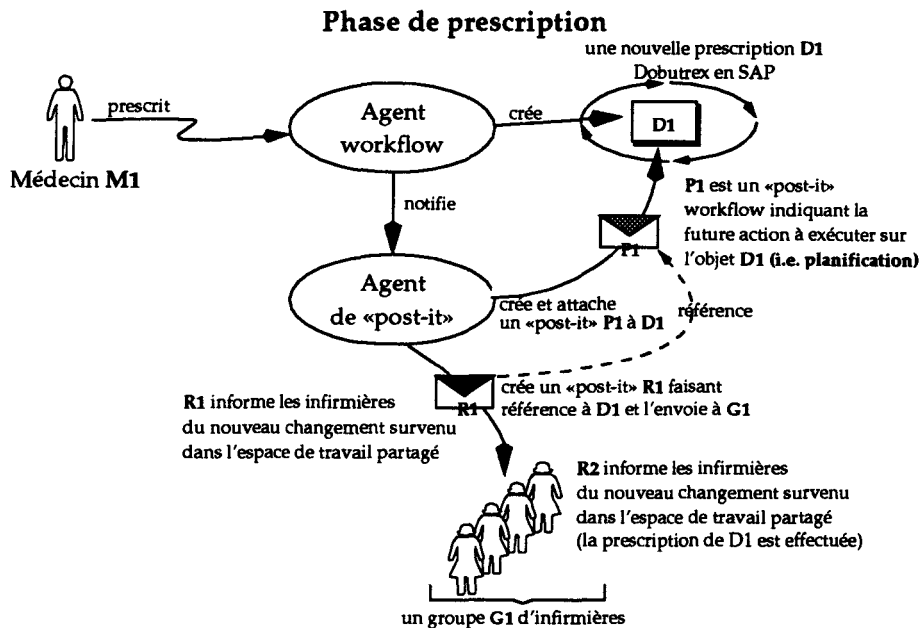


FIGURE 32 Un exemple d'un scénario de prescription d'un traitement D1: *Dobutrex en SAP*. Un «post-it» workflow P1 créé et attaché à la nouvelle prescription avec un «post-it» de coordination R1 envoyé aux destinataires susceptibles d'exécuter la future action, qu'est la planification, sur cet objet.

2. Planification: la figure 33 présente un scénario de planification de l'objet prescrit D1. L'acteur effectif I1, à qui incombe la tâche de planification de D1, prend connaissance de la tâche en consultant le «post-it» de coordination R1. Il planifie, via l'agent de workflow, le traitement D1. Ce nouveau changement est notifié à l'agent de «post-it» par l'intermédiaire de l'agent de workflow. Le premier procède donc à la création de deux «post-it» P2 et R2 qu'il attache respectivement à l'objet D1 et au groupe d'infirmières G1 pour les informer de la prochaine action à exécuter qui est la dispensation de D1.

Le «post-it» R1 fait référence à l'objet D1 et permet à l'acteur chargé de la dispensation de s'informer directement et facilement de l'action suivante à effectuer sur D1. Ce qui assure un meilleur séquençement des tâches et, par conséquent, garantit une coordination efficace des tâches.

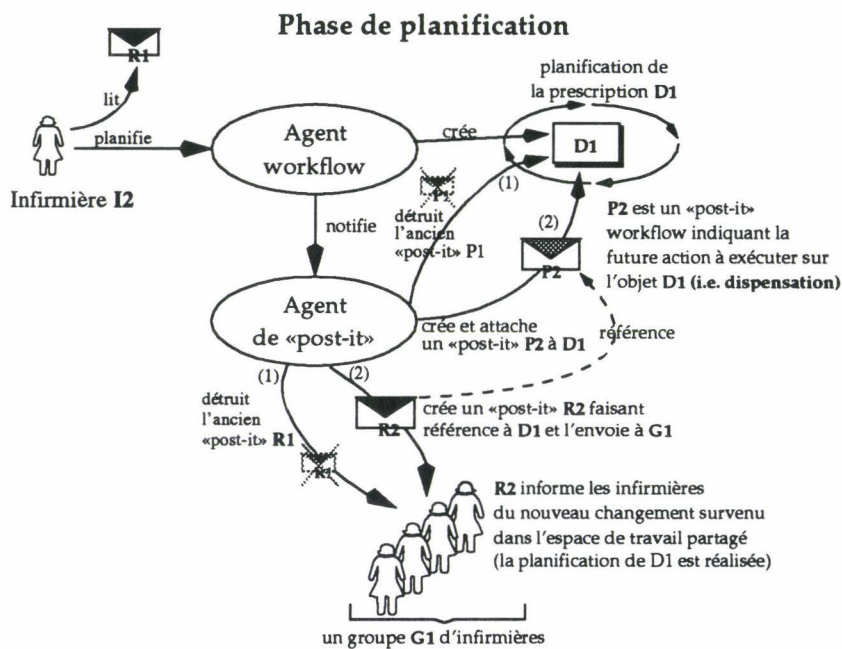


FIGURE 33

Un exemple d'un scénario de planification d'un objet prescrit D1. Un «post-it» workflow P2 est créé et est attaché à D1 avec un «post-it» de coordination R2 envoyé aux acteurs susceptibles d'exécuter la future action sur cet objet. Avant la création de P2 et de R2, le gestionnaire de «post-it» détruit les anciens «post-it» qui sont considérés obsolètes (P1 et R1) étant donné que l'action de planification qui leur est associée a été effectivement réalisée.

3. **Dispensation:** la figure 34 présente un scénario de dispensation de l'objet prescrit D1. L'acteur effectif I2, chargé de l'administration de D1, prend connaissance de cette tâche en consultant le «post-it» de coordination R2. Il dispense, via l'agent de workflow, le traitement D1 au patient. Ce nouveau changement d'état est notifié à l'agent de «post-it». Il procède donc à la création de deux «post-it» P3 et R3 qu'il attache respectivement à l'objet D1 et au groupe de médecins G2 pour les informer de la prochaine action à exécuter qui est l'évaluation médicale du traitement D1. Le «post-it» R3 fait référence à l'objet D1 et permet à l'acteur chargé de l'évaluation de s'informer directement et facilement de l'action suivante à effectuer sur D1. Les «post-it» P2 et R2 sont détruits aussitôt que l'action d'administration a été réalisée. Cette technique permet d'éviter de surcharger inutilement les objets du domaine en «post-it».

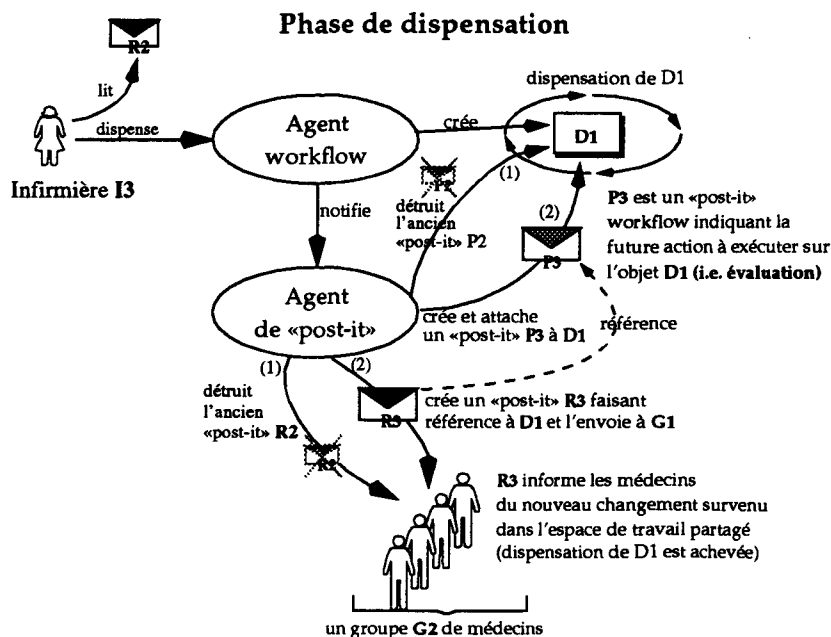


FIGURE 34 Un exemple d'un scénario de dispensation d'un objet prescrit D1. Un «post-it» workflow P3 est créé et est attaché à D1 avec un «post-it» de coordination R3 envoyé aux acteurs susceptibles d'exécuter la future action sur cet objet. Avant la création de P3 et de R3, le gestionnaire de «post-it» détruit les anciens «post-it» qui sont considérés obsolètes (P2 et R2) étant donné que l'action de dispensation qui leur est associée a été effectuée.

4. **Evaluation:** le scénario d'évaluation de l'objet prescrit D1 se présente de la même façon que les précédents. Comme le montre la figure 35, l'acteur effectif M2, chargé de l'éva-

luation de D1, prend connaissance de cette tâche en consultant le «post-it» de coordination R3 créé dans la phase de dispensation. Il évalue le traitement D1 administré au patient et modifie l'état de l'objet D1. Ce nouveau changement d'état est notifié à l'agent de «post-it». Celui-ci en crée deux P4 et R4, qu'il attache respectivement à l'objet D1 et au groupe de médecins G3 pour les informer de la fin du cycle de workflow pour D1 et de la décision médicale prise à l'égard de ce traitement qui est D1. Le «post-it» de coordination R4 fait référence à l'objet D1, et permet à l'acteur, chargé de l'évaluation, de s'informer directement et facilement de l'action suivante à effectuer sur D1. Avant de créer les deux nouveaux «post-it», l'agent de workflow détruit les anciens «post-it» P3 et R3 qui sont désormais obsolètes étant donné que D1 a été effectivement évalué.

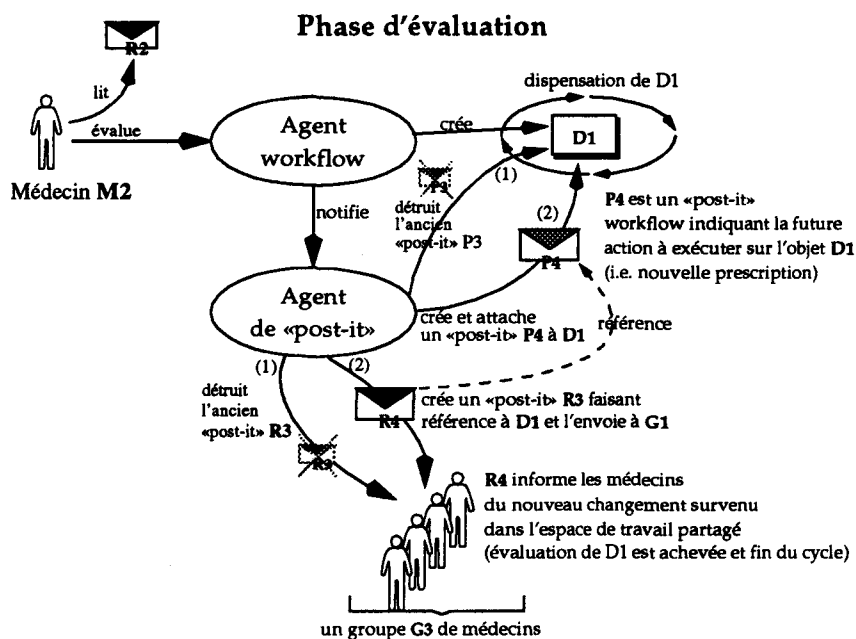


FIGURE 35

Un exemple d'un scénario d'évaluation d'un objet prescrit D1. Un «post-it» workflow P4 est créé et est attaché à D1 avec un «post-it» de coordination R4 envoyé aux acteurs susceptibles d'exécuter la future action sur cet objet. Avant la création de P4 et de R4, le gestionnaire de «post-it» détruit les anciens «post-it» qui sont considérés obsolètes (P3 et R3) étant donné que l'action d'évaluation qui leur est associée a été effectuée.

6.3 Conclusion

La modélisation des interactions entre les différents acteurs de soins par le modèle de workflow permet d'envisager la médiation de l'information par un système informatique. Ce dernier, par ses capacités de calcul et de communication, permet d'introduire un contrôle sur l'information (cohérence, disponibilité, complétude, synchronisation, etc.) et ainsi de diminuer, voire de supprimer les «breakdown».

La prescription en unité de soins intensifs est un domaine lourd de contraintes:

- contraintes qui sont liées aux données: nous évoquerons seulement la complexité de la représentation de connaissance des médicaments,
- contraintes qui sont liées aux acteurs: diversité, multiplicité et interchangeabilité (partielle) du personnel soignant,
- contraintes qui sont liées au temps réel et à la notion d'urgence plus ou moins forte.

L'utilisation d'agents nous permet en partie de faire face à ces contraintes [Yous95a]. Les agents permettent d'automatiser facilement de nombreuses tâches comme l'envoi automatique de «post-it» suite à certains événements, comme l'élaboration de la prescription à partir des données fournies par l'acteur ou encore comme la sélection des actions que doit réaliser en priorité un acteur à un instant donné.

L'emploi d'agents permet également de gérer assez simplement les problèmes liés à la substitution des acteurs, les uns par les autres. Ceci tout d'abord, parce qu'il assure la continuité de la liaison entre l'acteur et ses groupes d'appartenance (groupe thérapeutique de tel malade, groupe infirmier, groupe médical, etc.), et ensuite, parce qu'il permet d'obtenir pour l'acteur les droits lui permettant d'exécuter l'action souhaitée.

*Réalisation et
Evaluation du
Prototype de PLACO*

Notre propos, dans ce chapitre, n'est pas de présenter un outil informatique opérationnel prêt à être installé dans les USI. Nous voulons simplement valider le modèle PLACO, basé sur l'approche AWA de WINOGRAD & FLORES [ATI94], et montrer l'intérêt que peut apporter un système dédié à la coordination des tâches dans des organisations telles que les USI.

Nous nous consacrerons donc, dans ce chapitre, à la description de la réalisation, en langage VisualSmalltalk[®], d'un prototype de la solution PLACO que nous avons présentée dans le chapitre précédent. Nous verrons que ce prototype, de par son aspect évolutif, se prête également bien à des applications autres que la planification coopérative en unités de soins intensifs.

Ce chapitre se divisera en trois parties.

- Dans la première partie, nous décrirons les objets manipulés ainsi que leur composition. Pour ce faire, nous utiliserons une méthode de conception orientée objets connue sous le nom de OOD [Booc94a]. Nous détaillerons, par la même occasion, le graphe d'héritage des classes de notre solution.
- La deuxième partie sera consacrée à la description du prototype réalisé. Nous commencerons cette partie par une introduction au langage Smalltalk qui décrit ses fonctionnalités et ses caractéristiques. Ensuite, nous décrirons les aspects techniques du prototype et les problèmes soulevés.

- La troisième partie traitera de la validation du modèle et l'évaluation du prototype que nous avons développé. Nous décrirons les résultats de l'évaluation des fonctions du prototype et montrerons comment les acteurs de soins peuvent coopérer efficacement et, par conséquent, prévenir les erreurs de fonctionnement et les «breakdown» qui surgissent. Nous concluerons cette partie en décrivant, en terme de perspectives à court-terme, les améliorations que nous pouvons apporter à notre prototype.

7.1 La Représentation Orientée Objets de la Solution PLACO

Nous avons identifié dans le modèle un cycle conversationnel principal qui régit l'activité dite de planification coopérative en USI. Autour de ce cycle principal, un réseau de cycles secondaires de délégation et de clarification se greffe. La question qu'on tentera d'élucider dans ce paragraphe est de savoir *pourquoi et comment on peut utiliser l'approche orientée objets pour implanter le modèle PLACO?*

Pour ce faire, nous commencerons, d'abord, par expliquer en quoi l'approche orientée objets peut aider à l'implantation du modèle PLACO. Nous donnerons un aperçu général sur l'approche orientée objets et sur les avantages qu'elle offre quant à l'implantation physique d'un tel modèle. Par la suite, nous donnerons la description détaillée des différentes classes d'objets existant dans PLACO. Nous détaillerons, à cet effet, le graphe d'héritage des objets du domaine, de contrôle et de coordination, ainsi que leurs liens de composition.

7.1.1 Pourquoi l'approche orientée objets?

Les logiciels sont par nature complexes [Booc94a]; leur complexité surpasse souvent les capacités intellectuelles humaines. Comme le suggère BROOKS, «*la complexité des logiciels est une caractéristique innée et non accidentelle*» [Booc94b]. En effet, la complexité est inhérente aux problèmes à résoudre, à la difficulté de gérer le processus de développement et à la flexibilité qu'autorisent les logiciels. La complexité prend souvent la forme d'une hiérarchie dans laquelle un système se décompose en sous-systèmes reliés entre eux et ayant, à leur tour, leurs propres sous-systèmes. Il existe des raisons fondamentales et parfois incontournables d'appliquer la décomposition orientée objets, dans laquelle nous

interprétons le monde comme un ensemble d'objets qui collaborent pour réaliser un certain comportement de plus haut niveau.

La conception orientée objets constitue la méthode qui nous conduit à la décomposition orientée objets. Elle offre un riche ensemble de modèles physiques et logiques avec lesquels nous pouvons raisonner au sujet des différents aspects du système considéré. Or, notre système PLACO est considéré complexe du fait qu'il intègre différents types d'information.

Dans les méthodes de conception, l'accent est mis sur la structuration appropriée et efficace d'un système complexe. Mais qu'est ce que la conception orientée objets? D'après Grady BOOCH, *«la conception orientée objets (COO) est une méthode de conception incorporant le processus de décomposition orienté objets et une notation permettant de dépendre à la fois les modèles logiques et physiques, aussi bien que statiques et dynamiques, du système à concevoir»* [Booc94b]. Cette méthode de conception utilise des abstractions de classes et d'objets pour structurer logiquement les systèmes, alors que la conception structurée prône des abstractions algorithmiques.

L'approche orientée objets présente l'avantage de structurer un système en objets facilement interprétables. Elle nous aide à mieux organiser la complexité inhérente aux logiciels, de même qu'elle nous aide à décrire la complexité organisée des systèmes. La décomposition orientée objets offre un certain nombre d'avantages hautement significatifs sur la décomposition algorithmique classique. Elle génère de plus des petits systèmes, grâce à la réutilisation de mécanismes communs, procurant ainsi une importante économie d'expression. Les systèmes orientés objets réagissent mieux aux changements, et par conséquent, sont plus à même d'évoluer avec le temps, car leur architecture est basée sur des formes intermédiaires stables. En réalité, elle diminue fortement le risque de la construction de logiciels complexes, car ils sont alors conçus pour évoluer de façon progressive à partir de systèmes plus petits et plus fiables. Par ailleurs, la décomposition orientée objets s'attaque à la complexité inhérente aux programmes en nous aidant à prendre des décisions. C'est pour ces raisons que notre choix s'est porté sur l'approche orientée objets pour la conception et l'implémentation du système PLACO.

7.1.2 La représentation orientée objets de PLACO

La décomposition d'un système complexe peut conduire à de larges ensembles d'objets. Pour gérer ces objets de manière simple, il convient de déterminer les relations qui existent entre ces objets. Certaines de ces relations peuvent être abordées avec la notion de hiérarchie qui permet d'assurer un ordre à l'ensemble des abstractions obtenues. La classification est le moyen par lequel nous ordonnons nos connaissances. En COO, la reconnaissance de similitudes entre «choses» nous permet d'exposer les abstractions et mécanismes clefs, et finalement nous conduit à des architectures plus réduites et plus simples [Booc94a]. Nous proposons, dans ce qui suit, d'utiliser la méthode de conception orientée objets OOD de Grady BOOCH [Booc94a] pour représenter le système PLACO. Mais avant, nous donnerons une vision globale de ce qu'est la méthode OOD.

7.1.2.1 Principes de la méthode OOD

En 1983, Grady BOOCH proposait une méthode de conception dite orientée objets. En fait, cette méthode était largement dédiée au langage ADA et ne permettait pas de prendre en compte l'héritage et le polymorphisme. Une notation graphique simple permettait de représenter les «packages», leurs interfaces, les tâches et les sous-programmes. Cette méthode adoptait une démarche en cinq étapes:

- identifier les objets,
- identifier les opérations,
- établir la visibilité,
- établir les interfaces,
- implémenter les objets.

Une deuxième version de cette méthode a été présentée fin 1990. Conscient de l'impact des langages à objets tels que Smalltalk, C++ ou Eiffel, BOOCH propose une méthode indépendante d'ADA, utilisant les concepts de classe, d'héritage, d'envoi de message, ainsi que toutes sortes de relations entre classes. Les nouvelles étapes principales sont au nombre de quatre [Hill93]:

1. Identifier les classes et les objets à un niveau d'abstraction donné. Le processus de conception débute avec la recherche des classes et des objets propres au vocabulaire du domaine, tout en se restreignant au problème que l'on souhaite résoudre.

2. Identifier la sémantique de ces classes et de ces objets. Le concepteur doit détailler la représentation interne des classes de manière à comprendre leur fonctionnement et leur rôle précis. Ceci permet de déterminer les interfaces de chaque classe.
3. Identifier les relations existant entre les classes et les objets.
4. Implémenter ces classes et ces objets. Si le niveau d'abstraction utilisé précédemment est trop élevé, il peut être nécessaire d'itérer certaines étapes afin d'obtenir un niveau de détail suffisant pour la programmation.

La méthode de conception OOD, utilisée en aval d'une méthode d'analyse, permet de donner une représentation claire et une description précise du comportement des objets d'une architecture logicielle donnée. Nous nous proposons donc d'utiliser la méthode OOD pour représenter les différentes classes et différents objets de notre système PLACO.

7.1.2.2 Représentation des objets PLACO selon OOD

Nous proposons la décomposition de la solution PLACO en quatre classes d'objets principales qui sont *ObjetPersonne*, *ObjetPrescrit*, *WorkflowItem* et *PostIt*.

Le diagramme de la figure 36 montre l'existence des classes principales et leurs relations. Il indique également les rôles et les responsabilités communs aux entités qui participent au comportement du système PLACO.

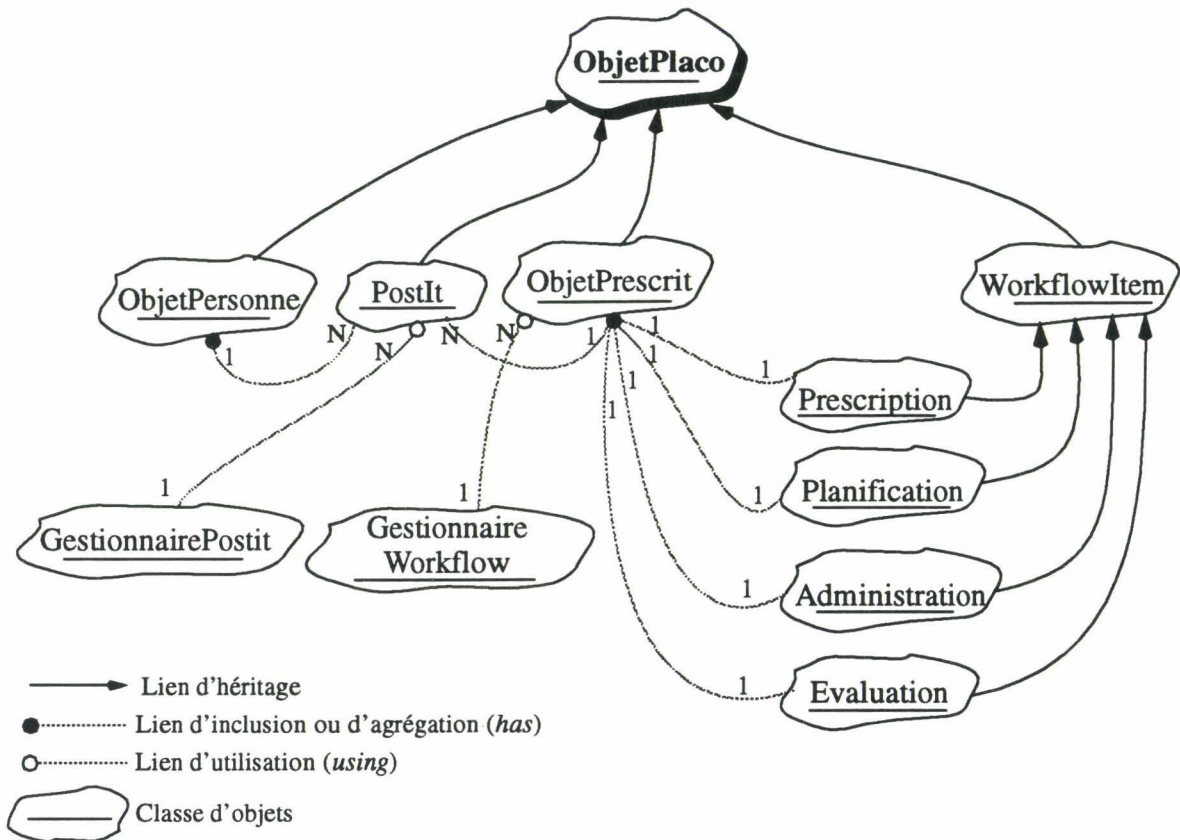


FIGURE 36 Diagramme des classes principales du système PLACO. Il se compose de quatre classes: la classe **ObjetPersonne**, la classe **ObjetPrescrit**, la classe **WorkflowItem** et la classe **PostIt**.

1. **ObjetPersonne** représente la classe des acteurs de soins: **ActeurSoins** et celle des patients: **Malade**. Les acteurs de soins se répartissent soit en groupes de personnes (groupe d'infirmières: **GrpInfirmieres**, groupe de médecins: **GrpMedecins** et groupe mixte: **GrpMixte**), soit en individus (infirmière: **Infirmiere** et médecin: **Medecin**).
2. **ObjetPrescrit** représente la classe des médicaments: **Medicament**, des actes médicaux: **ActeMedical** et des examens médicaux: **ExamenMedical**.
3. **WorkflowItem** représente les phases du cycle de workflow. Il se décompose donc en quatre entités telles que la prescription: **Prescription**, la planification: **Planification**, l'administration: **Administration** et l'évaluation: **Evaluation**.

4. **PostIt** représente le message. Cette classe contiendra les «post-it» de coordination (attachés aux acteurs), les «post-it» de workflow (attachés aux objets du domaine) et les «post-it» libres.

Spécification des classes

Dans PLACO, les classes ne sont pas isolées et collaborent entre elles. Les connexions principales comprennent les relations d'héritage, d'utilisation («*using*») et d'inclusion ou d'agrégation («*has*»).

1. Objets prescrits

Ainsi, comme l'illustre la figure 36, un objet de la classe **ObjetPrescrit** contient:

- un objet de prescription de la classe **Prescription**,
- un objet de planification de la classe **Planification**,
- un objet d'administration de la classe **Administration**,
- un objet d'évaluation de la classe **Evaluation**.

Un objet de la classe **ObjetPrescrit** peut être un médicament, un acte médical ou un examen médical. Par conséquent, la classe **ObjetPrescrit** se décomposera en classe **Medicament**, classe **ExamenMedical** et classe **ActeMedical**. Ces trois sous-classes représentent les objets qu'on peut prescrire en USI.

Remarque 1: Il est clair que dans le domaine médical, les types de prescription, en général, sont divers. Cependant, nous nous sommes limités, dans notre prototype, aux trois catégories les plus courantes, à savoir un médicament, un examen médical et un acte médical.

Les objets prescrits sont gérés par un agent qui s'appelle *agent de workflow*. A cet effet, la classe **ObjetPrescrit** utilisera une classe appelée **GestionnaireWorkflow** dédiée à la gestion des états des objets prescrits dans le workflow.

Etant donné que tout objet prescrit et tout acteur possèdent des «post-it», les objets des classes **ObjetPrescrit** et **ObjetPersonne** possèdent des objets de la classe **PostIt**.

2. les «post-it»

Comme les «post-it» se divisent en trois grandes catégories, «post-it» de workflow, «post-it» de coordination et «post-it» libre, nous proposons de subdiviser la classe **PostIt** en trois

sous-classes correspondant respectivement à **PostitWorkflow**, **PostitCoordination** et **PostitFree**.

La classe **PostitWorkflow** correspond à la classe des «post-it» attachés aux objets de la classe **ObjetPersonne**.

La classe **PostitCoordination** correspond à la classe des «post-it» attachés aux acteurs, donc aux objets de la classe **ObjetPersonne**.

La classe **PostitFree** correspond à la classe des «post-it» libres attachés aux acteurs, donc aux objets de la classe **ObjetPersonne**.

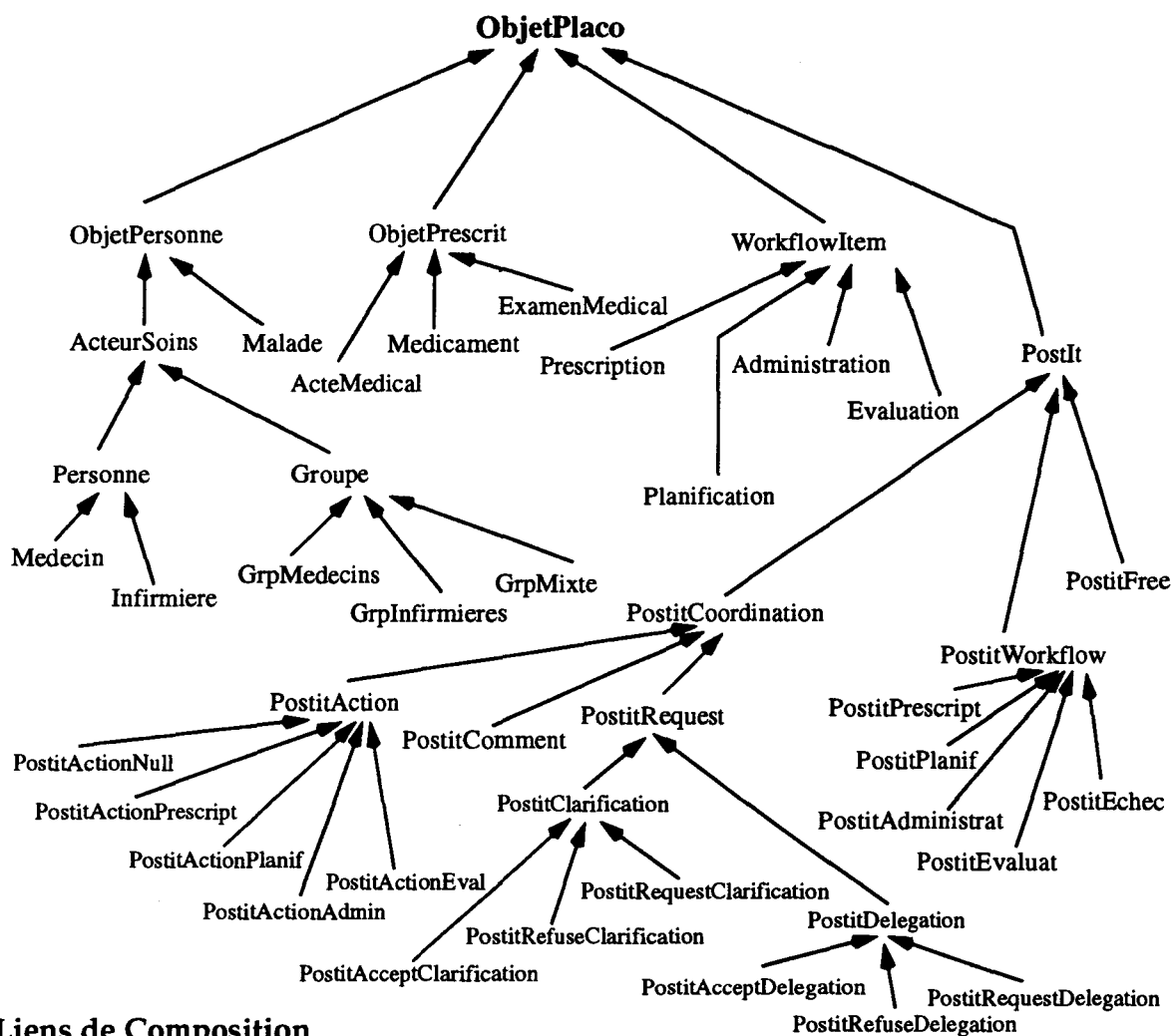
Un «post-it» de coordination peut se présenter sous différentes formes selon son contenu. Il peut concerner l'exécution d'une action, une demande de clarification ou de délégation, ou un simple commentaire lié à l'exécution d'une action. Ainsi, nous distinguons dans la classe **PostitCoordination** trois types de «post-it» correspondant à trois sous-classes:

1. La classe **PostitAction**.
2. La classe **PostitRequest**.
3. La classe **PostitComment**.

Par ailleurs, la classe **PostIt** utilise la classe **GestionnairePostit**, cette dernière étant dédiée à la gestion des «post-it» et à leur mise à jour. Elle renferme toutes les méthodes exécutables sur les «post-it» et nécessaires à leur gestion et mise à jour.

La décomposition globale en classes d'objets de la solution que nous proposons peut être définie comme le montre la figure 37 (voir également l'annexe A pour la liste des classes définies à partir de la classe **ObjetPlaco**).

Graphe d'Héritage des classes PLACO



Liens de Composition

- Malade → Ensemble des objets «ObjetPrescrit», Ensemble des objets «PostIt»,
- ObjetPrescrit → un objet «Prescription»
- ObjetPrescrit → un objet «Planification»
- ObjetPrescrit → un objet «Dispensation»
- ObjetPrescrit → un objet «Evaluation»
- ObjetPrescrit → Ensemble des objets «PostIt»,

FIGURE 37 Graphe d'héritage des classes d'objets de PLACO avec leurs liens de composition.

7.2 Réalisation du Prototype de PLACO

7.2.1 Le langage Smalltalk

L'utilisation de la technologie orientée objets s'applique à un large spectre de langages de programmation basés sur objets et orientés objets. Quelle que soit l'importance des méthodes d'analyse et de conception, il est impossible d'ignorer les détails de l'écriture du code car, en fin de compte, un langage de programmation est:

- un outil de conception,
- une denrée destinée à la consommation humaine [Booc94a],
- un moyen permettant de donner des instructions à un ordinateur.

7.2.1.1 Contexte et historique:

Smalltalk a été créé par les membres du groupe de recherches sur l'apprentissage du Centre de Recherche Xerox de Palo Alto. Ce langage a été principalement influencé par SIMULA, bien qu'il ait également emprunté quelques idées au langage FLEX et aux travaux de Seymore PAPERT et Wallace FEURZEIG. Smalltalk représente à la fois un langage et un environnement de développement logiciel. C'est un langage de programmation orienté objets «pur», en ce sens que tout y est objet, même les entiers et les classes. Après Simula, Smalltalk est peut-être le langage de programmation orienté objets le plus important. Ses concepts ont influencé non seulement la conception de presque tous les autres langages de programmation orientés objets, mais aussi l'aspect et la convivialité des interfaces-utilisateurs graphiques telles que l'interface-utilisateur de Macintosh, de Windows et de Motif.

Smalltalk a nécessité près d'une décennie de travail, et a été le résultat de la synergie au sein d'un groupe. Il existe actuellement cinq versions identifiables de Smalltalk, distinguées par leur année d'apparition: Smalltalk-72, -74, -76, -78 et l'incarnation la plus courante, Smalltalk-80.

Smalltalk/V, un dialecte important de Smalltalk, très proche de Smalltalk-80, est diffusé par Digitalk Inc. Excepté pour les classes de l'interface-utilisateur, les bibliothèques de classes sont très proches. Tout comme pour Smalltalk-80, l'environnement et les outils de

développement offrent les mêmes possibilités, même si leurs structures et fonctionnalités diffèrent.

7.2.1.2 Présentation

La citation suivante de INGALLS [Inga76], «*Le but du projet Smalltalk est de guider les enfants à tous âges dans le monde de l'information. Le défi consiste à identifier et à maîtriser des métaphores d'une simplicité et d'une puissance suffisantes pour permettre à une personne d'avoir accès à des informations et de les contrôler de façon créative. Ces informations peuvent aller des nombres et du texte jusqu'aux sons et images*», permet de mettre en évidence les idées qui ont motivé la conception du langage Smalltalk.

Dans ce but, Smalltalk est bâti autour de deux concepts simples: tout est traité comme un objet, et les objets communiquent en échangeant des messages.

Dans le tableau 10, nous résumons les propriétés de Smalltalk. Bien que le tableau ne l'indique pas, l'héritage multiple est possible en Smalltalk, au moyen de la redéfinition de certaines méthodes primitives [Booc94b].

Abstraction	Variables d'instances	Oui
	Méthodes d'instances	Oui
	Variables de classes	Oui
	Méthode de classes	Oui
Encapsulation	Des variables	Privée
	Des méthodes	Publique
Modularité	Genres de modules	Aucun
Hiérarchie	Héritage	Simple
	Unités génériques	Non
	Métaclases	Oui
Typage	Fortement typé	Non
	Polymorphisme	Simple
Simultanéité	Multi-tâches	Indirecte (par les classes)
Persistance	Objets persistants	Non

TABEAU 10 Tableau récapitulatif des propriétés du langage Smalltalk.

Une des plus grandes qualités du langage Smalltalk est la facilité de prototypage. Il offre la possibilité de créer facilement des interfaces graphiques de type **Windows**[®]. En outre, il permet aux concepteurs de développer leurs applications de façon incrémentale. Le test partiel des applications en cours de développement est un des plus grands atouts de Smalltalk. Il est donc particulièrement adapté au prototypage des logiciels complexes.

Le langage utilisé dans le prototypage de PLACO est **VisualSmalltalk**[®]/V3.0 de Digitalk Inc., qui est dérivé de Smalltalk-80 et qui fonctionne sous **Windows**[®].

7.2.2 Description du prototype de PLACO

Le prototype que nous avons réalisé est une application mono-utilisateur qui tourne sous **Windows for Workgroups**[®]. L'objectif que nous nous sommes fixé, dans un premier temps, est de valider le modèle proposé ainsi que de montrer l'apport d'un système de coordination de tâches vis-à-vis de l'organisation des activités au sein des unités de travail.

7.2.2.1 Création d'un objet prescrit

Dans le prototype PLACO, les étapes de création d'un *objet prescrit* sont décrites de la façon suivante:

- création d'une instance vide de *ObjetPrescrit*,
- saisie du type de prescription à faire. Une prescription peut être de type **Médicament** (e.g. **Dobutrex** en SAP), de type **Acte Médical** (e.g. surveillance des paramètres) ou de type **Examen Médical** (e.g. analyse du sang),
- création effective de la prescription (remplir le nom de l'objet prescrit, l'information relative au dosage, à la fréquence de prise par jour, à la durée de prise, etc.) pour l'objet prescrit, selon le type de prescription choisie auparavant (i.e. **Medicament**, **ActeMedical** ou **ExamenMedical**). L'objet de prescription est une instance de la classe **Prescription**, qui est sous-classe de **WorkflowItem**,
- création de la planification pour l'objet prescrit (vérifier la faisabilité de cette action, définir les horaires exacts de prise, etc.). L'objet de planification est une instance de la classe **Planification**, qui est sous-classe de **WorkflowItem**,
- création de l'administration (i.e. dispensation) pour l'objet prescrit (dispensation de la prescription au patient en respectant les horaires de prise fixés auparavant). L'objet

d'administration est une instance de la classe **Administration**, qui est sous-classe de **WorkflowItem**.

- création de l'évaluation pour l'objet prescrit (évaluation clinique des résultats obtenus après l'administration de la prescription au patient). L'objet d'évaluation est une instance de la classe **Evaluation**, qui est sous-classe de **WorkflowItem**.

La création d'un objet prescrit se fait à partir de l'objet **MALADE** (il s'agit d'une méthode de la classe **MALADE**). On crée une instance vide de la classe **ObjetPrescrit** que les acteurs de soins rempliront au fur et à mesure de l'évolution de celle-ci dans le cycle de workflow (prescription, planification, administration et enfin, évaluation).

Dans chaque phase, les acteurs enrichissent les connaissances de l'objet prescrit. Ainsi, pour chaque objet prescrit, on crée quatre objets correspondant aux quatre phases du cycle de workflow. On aura alors le schéma suivant:

1. Dans la phase de prescription, on crée un objet de prescription qui est une instance de la classe **Prescription**. Cet objet est renseigné et attaché à l'objet prescrit en question par un lien de composition.
2. Lors de la phase de planification, on crée un objet de planification qui est une instance de la classe **Planification**. De même, cet objet est renseigné et attaché à l'objet prescrit par un lien de composition.
3. Durant la phase d'administration (ou de dispensation), on crée un objet d'administration qui est une instance de la classe **Administration**. Cet objet est renseigné et attaché à l'objet prescrit par un lien de composition.
4. Dans la phase d'évaluation, on crée un objet d'évaluation qui est une instance de la classe **Evaluation**. Cet objet est renseigné et attaché à l'objet prescrit en question.

7.2.2.2 Création d'un objet «post-it»

A l'issue de chaque phase, l'agent de «post-it» (i.e. **GestionnairePostit**) génère un premier «post-it», de type workflow, qu'il attache à l'objet prescrit (sujet à cette modification). Il crée également un deuxième «post-it», de type coordination, référençant l'objet prescrit, qu'il envoie au groupe d'infirmières et/ou de médecins. Ce «post-it» permet l'accès direct à la prescription en question, et donc au dossier du patient. Cette technique présente

l'intérêt de dispenser les utilisateurs de «balayer» toute la liste des dossiers de patients pour rechercher les actions à effectuer.

Concrètement, lors de création d'une prescription pour un objet prescrit vide, on génère:

- un objet «post-it» (de workflow) qui est une instance de la classe **PostitPrescript** (sous-classe de **PostitWorkflow**). Ce «post-it» est attaché automatiquement à l'objet prescrit pour indiquer son état dans le workflow,
- un objet «post-it» (de coordination) qui est une instance de la classe **PostitActionPrescript** (sous-classe de **PostitAction**, elle-même sous-classe de **PostitCoordination**). Ce «post-it» est attaché à un groupe de médecins pour leur indiquer le type d'action à exécuter (i.e. la prescription). Grâce aux liens de référence que contient ce «post-it», l'acteur destinataire peut accéder directement à l'objet prescrit afin d'exécuter l'action de prescription.

Comme illustré dans la figure 38, les «post-it» attachés aux acteurs sont, en réalité, rangés dans un espace mémoire partagé par l'ensemble des acteurs. Seules des références aux «post-it» sont envoyées aux destinataires.

Dès lors qu'un acteur exécute l'action à laquelle fait référence le «post-it», le gestionnaire de «post-it» détruit ce dernier du fait qu'il est considéré obsolète. Toute action réalisée effectivement entraîne systématiquement la destruction du «post-it» qui la référence.

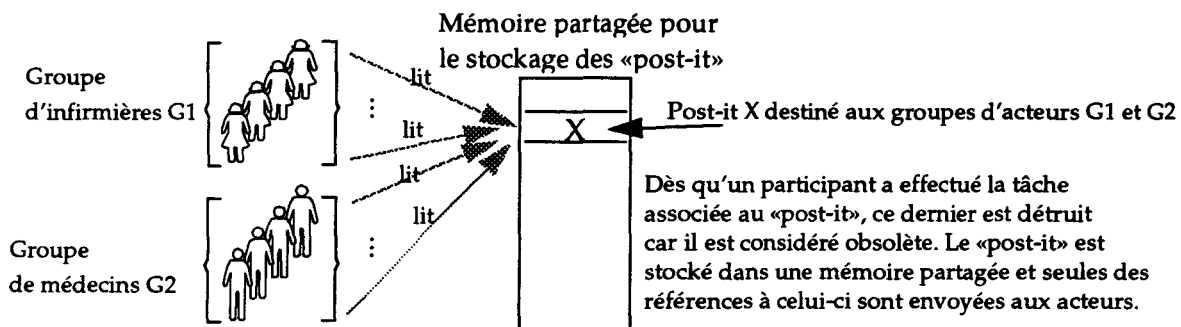


FIGURE 38 Utilisation de la mémoire partagée pour faciliter la gestion des «post-it».

Ainsi, l'utilisation de la mémoire partagée permet d'éviter la duplication des «post-it» pour chaque acteur (surcharge des boîtes de messages). Elle permet également de faciliter la mise à jour de ces derniers (modification, destruction, etc.).

7.2.2.3 Création d'une délégation d'une tâche

Pour déléguer une tâche, un acteur X doit s'adresser à un autre acteur Y en créant un *sous-cycle de délégation*. Ce sous-cycle est contrôlé par l'envoi de «post-it» d'un des acteurs vers l'autre. A cet effet, un objet «post-it», instance de la classe **PostitRequestDelegation** qui est sous-classe de **PostitDelegation**, est émis par l'acteur X vers l'acteur Y. Ce «post-it» contient toute l'information relative à la délégation de la tâche en cours (par exemple, la prescription, la planification, etc.). Ce «post-it» possède une durée de vie ΔT qui est limitée et fixée par l'acteur demandeur.

Plusieurs cas de figure se présentent:

1. Si l'acteur destinataire ne répond pas, alors, à l'issue d'un laps de temps ΔT , le gestionnaire de «post-it» renvoie une *réponse négative* à l'acteur demandeur par la création d'un «post-it» qui est une instance de la classe **PostitRefuseDelegation** (sous-classe de **PostitDelegation**).
2. Si l'acteur Y répond à la demande de délégation, il peut envoyer une *réponse négative* pour signaler un refus de la délégation, ou *positive* pour signaler l'acceptation de la délégation. En cas de refus, nous retrouvons le même scénario que précédemment, c'est-à-dire que le gestionnaire de «post-it» renvoie une *réponse négative* à l'acteur demandeur au moyen d'un «post-it», qui est une instance de la classe **PostitRefuseDelegation**. En cas d'acceptation de la délégation, le gestionnaire de «post-it» crée un nouveau «post-it», qui est une instance de la classe **PostitAcceptDelegation** (sous-classe de **PostitDelegation**), et l'envoie à l'acteur X. Ce «post-it» est d'abord signé par l'acteur Y.

Remarque 2: Dans le cas où la demande de délégation est acceptée, elle prend fin dès l'envoi du «post-it» de réponse par l'acteur Y.

En revanche, en cas de refus de délégation, l'acteur demandeur X possède toute la latitude de propager sa demande à d'autres acteurs. On aura alors le même scénario.

Remarque 3: Dans les deux cas de figure, le gestionnaire de «post-it» crée un «post-it» workflow de type **PostitWorkflow** qu'il attachera à l'objet prescrit sur lequel la tâche déléguée va être exécutée. Ce «post-it» rend compte de l'état de l'objet prescrit dans le

workflow, parce qu'il signale qu'une délégation a été faite et qu'elle a été refusée ou acceptée.

7.2.2.4 Création d'une clarification d'une tâche

La clarification d'une tâche se passe entre deux acteurs **X** et **Y**. Elle fait l'objet d'une création d'un *sous-cycle de clarification*. Ce sous-cycle est contrôlé par l'envoi de «post-it» de part et d'autre des deux acteurs. A cet effet, un objet «post-it», instance de la classe **PostitRequestClarification**, qui est sous-classe de **PostitClarification**, est émis par l'acteur **X** vers l'acteur **Y** contenant une demande ou une offre de clarification de la tâche en cours (par exemple, la prescription, la planification, l'administration ou l'évaluation). Ce «post-it» possède une durée de vie ΔT qui est limitée et fixée par l'acteur demandeur.

Comme dans le cas de la délégation, plusieurs scénarios se présentent:

1. Si l'acteur destinataire ne répond pas, alors, à l'issue du délai ΔT , le gestionnaire de «post-it» renvoie une *réponse négative* à l'acteur demandeur au moyen d'un «post-it», qui est une instance de la classe **PostitRefuseClarification** (sous-classe de **PostitClarification**).
2. Si l'acteur **Y** répond à la demande, il peut envoyer une *réponse négative* pour signaler un refus de la clarification, ou *positive* pour signaler l'acceptation de la clarification. Dans le premier cas, nous retrouvons le même cas de figure expliqué précédemment. Le gestionnaire de «post-it» renvoie donc une *réponse négative* à l'acteur **X** au moyen d'un «post-it», qui est une instance de la classe **PostitRefuseClarification**. Dans le deuxième cas, c'est-à-dire en cas d'acceptation, le gestionnaire de «post-it» crée un «post-it», qui est une instance de la classe **PostitAcceptClarification**, qui est elle-même sous-classe de **PostitClarification**, et l'envoie à l'acteur **X**. Ce «post-it» est signé d'abord par l'acteur **Y**.

En cas d'acceptation, le sous-cycle de clarification prend fin à l'issue de l'envoi du «post-it» réponse.

En revanche, en cas de refus de clarification, l'acteur demandeur possède toute la latitude de propager sa demande à d'autres acteurs. On aura alors le même scénario décrit précédemment.

Remarque 4: Dans les deux cas, le gestionnaire de «post-it» crée un «post-it» workflow de type **PostitWorkflow** qu'il attachera à l'objet prescrit sur lequel la tâche clarifiée va être exécutée. Ce «post-it» permet de rendre compte de l'état de l'objet prescrit dans le workflow en signalant qu'une clarification a été faite et qu'elle a été refusée ou acceptée.

7.2.2.5 Génération d'un échec lors de l'exécution d'une action

Lorsqu'un acteur se rend compte qu'une action quelconque ne peut pas être exécutée pour une raison donnée, il peut générer un arrêt ou ce que nous appelons dans le prototype, un échec. Cet échec peut avoir plusieurs causes, par exemple l'indisponibilité de matériel ou de personnel, le manque d'informations nécessaires à l'accomplissement de la tâche, la rupture de stock d'un médicament en pharmacie, etc.

Dans ce cas, il est intéressant de pouvoir donner à un acteur la possibilité de générer explicitement un échec dans la réalisation de sa tâche. Le gestionnaire de «post-it» crée alors deux «post-it»:

1. Le premier est un «post-it» de type action nulle que ce gestionnaire envoie aux acteurs concernés. Ce «post-it» n'est autre qu'une instance de la classe *PostitActionNull*, qui est sous-classe de *PostitAction* et de *PostitCoordination*.
2. Le deuxième «post-it» est une instance de la classe *PostitEchec*, sous-classe de *PostitWorkflow*, que ce gestionnaire attache à l'objet prescrit. Ce «post-it» rend compte de l'état de cet objet dans le workflow et impose une évaluation de cet arrêt.

Quelle que soit la phase dans laquelle l'objet prescrit se trouve, le gestionnaire de workflow «court-circuite» toutes les phases du cycle de workflow non encore accomplies, et impose une action d'évaluation sur l'objet prescrit sujet à cet échec.

7.2.3 Aspects techniques du prototype

Le prototype que nous avons réalisé a suivi un processus de développement incrémental. Nous avons veillé, le long de la phase de développement, à séparer l'information du domaine, de celle du contrôle. Cette séparation a pour vocation de bien faire ressortir les aspects d'information, ceux de coordination et ceux du contrôle de workflow.

Pour ce faire, nous présenterons, dans cette section, deux aspects techniques importants. Ils permettent un développement évolutif et incrémental de notre application.

7.2.3.1 Séparation entre le contrôle et le domaine

L'application réalisée se compose de deux grandes parties qui sont implémentées séparément:

- La première partie représente le domaine d'information contenant des objets. Un objet d'information est un objet qui peut être **vide**, **prescrit**, **planifié**, **dispensé** ou **évalué**. Ces cinq étapes, détaillées précédemment, décrivent le cycle de vie d'un objet d'information. Chaque étape est réalisée par un acteur via une interface graphique.
- La deuxième partie représente les connaissances de contrôle. Elle compte trois agents: agent d'interface, agent de workflow et agent des «post-it»:
 1. L'agent d'interface joue le rôle d'intermédiaire entre l'acteur et les agents de contrôle et des «post-it». Il communique aux deux autres agents tous les événements créés par l'acteur qui manipule les objets d'information.
 2. L'agent de Workflow prend en charge le contrôle des transitions des objets prescrits, objets d'information qui circulent d'une phase du cycle de workflow à l'autre. Il permet d'autoriser les accès aux objets selon les droits des acteurs. Il a également la charge de notifier à l'agent des «post-it» les actions exécutées sur les objets.
 3. L'agent de «post-it» prend en charge la coordination des tâches entre les acteurs. Il crée et met à jour les «post-it» qui sont attachés, soit aux objets d'information, soit aux acteurs. La gestion des «post-it» suit deux règles simples. Premièrement, à chaque événement est associée la création d'un «post-it» de workflow indiquant l'état courant de l'objet, et d'un «post-it» de coordination indiquant l'action à exécuter sur cet objet. Deuxièmement, à l'issue de chaque action, une mise à jour des «post-it» existants est déclenchée.

Avantages:

La séparation entre le contrôle et le domaine présente, pour notre prototype, deux grands avantages. En effet, cette approche permet:

- d'avoir un **prototype évolutif**. Sans modifier les objets d'information, on peut rajouter aux agents des connaissances de contrôle supplémentaires et assurer le même

fonctionnement des «post-it». On peut également en créer d'autres facilement suivant le type d'application,

- la réutilisation du prototype pour un autre type d'application tels que l'agencement des réunions, le suivi et la gestion des projets, par exemple au sein de **Ganymede**, etc. Pour ce faire, il suffit de modifier les règles de déclenchement des agents ainsi que le contenu des «post-it».

7.2.3.2 Le modèle multi-agents MVC

Définition

Le modèle multi-agents de Smalltalk-80, connu sous le nom de MVC (Modèle, Vue, Contrôleur) [Gold84], se présente de la façon suivante (voir figure 39):

1. **Modèle**: contient les structures de données et les actions associées à l'information qui sera vue de l'extérieur.
2. **Vue**: est la représentation externe du **Modèle**. L'utilisateur perçoit l'information associée au modèle par l'entremise de la **Vue** et peut communiquer avec celle-ci pour réaliser certaines actions.
3. **Contrôleur**: prend en charge les communications établies entre l'utilisateur, via la souris et le clavier, et la **Vue**. Il informe le **Modèle** des interactions faites sur le **Vue**.

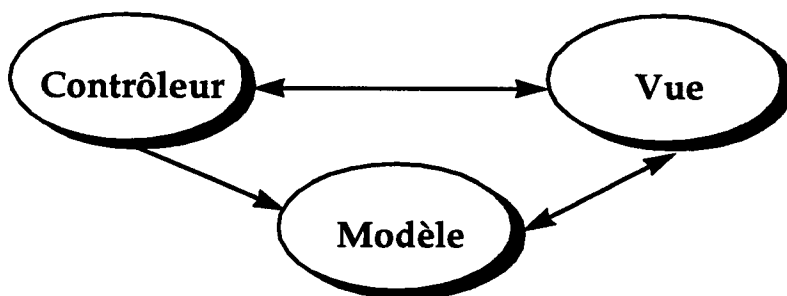


FIGURE 39 Le modèle MVC de SmallTalk-80 correspondant au **Modèle**, à la **Vue** et au **Contrôleur**.

Un **Modèle** peut être associé à plusieurs **Vues** et **Contrôleurs**. Par contre et pour des raisons de cohérence, une **Vue** ne peut être associée qu'à un seul **Modèle**.

La modularité de MVC permet «la réutilisabilité» des composants **Vue** et **Contrôleur** pour d'autres **Modèles**.

Avantages

La séparation entre l'interface et l'application permet de construire aisément plusieurs vues pour la même application, en définissant un mécanisme d'accès concurrent à l'application. Ce principe de séparation offre trois modes de distribution possibles [Dewa92]:

1. Distribution de l'application.
2. Distribution de l'interface-utilisateur.
3. Distribution des liens existant entre l'interface et l'application.

Application du modèle MVC au prototype PLACO

Pour des soucis de conformité au modèle MVC, nous avons choisi de construire une **Vue** du cycle de workflow principal qui communique avec le modèle **WorkflowItem** (prescription, planification, administration et évaluation), via le contrôleur du modèle.

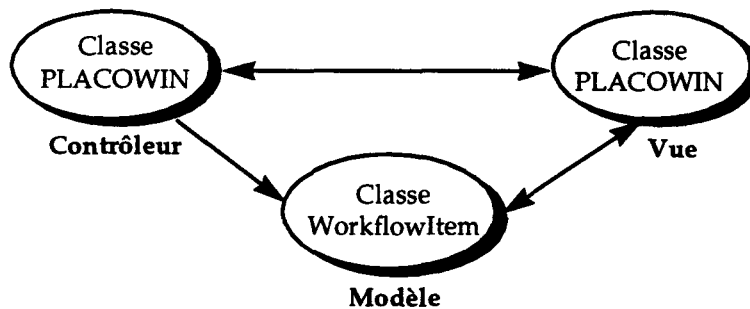


FIGURE 40 Application du modèle MVC de SmallTalk-80 à PLACO. La classe **PLACOWIN** sert de **Vue** et de **Contrôleur** pour le **Modèle** de la classe **WorkflowItem**.

D'après la figure 40, la classe **PLACOWIN** sert de **vue** (i.e. interface graphique) pour le modèle **WorkflowItem**. Cette interface représente les cycles du workflow principal, les menus déroulants et les actions qui s'exécutent sur les objets prescrits. Cette même classe sert de **contrôleur** pour le même modèle, et a pour rôle de traduire les interactions de l'utilisateur en les transmettant au modèle représenté par la classe **WorkflowItem**. Ce

modèle contient les structures de données et les méthodes adéquates et nécessaires pour l'exécution d'une action donnée. Le résultat de chaque opération est renvoyé par le modèle vers la vue pour l'affichage.

Afin d'illustrer cette approche, nous nous proposons de montrer les copies d'écran de deux interfaces graphiques que nous avons réalisées en respectant le modèle MVC. Toutefois, pour plus de détail sur le contenu des interfaces et les fonctionnalités qu'elles offrent, nous renvoyons le lecteur à l'annexe B.

L'écran 1 illustre l'interface graphique appelée **WorkflowItem** qui est propre à un patient. Elle contient la liste des objets prescrits, la liste des «post-it» de workflow relatifs à l'état de chaque objet prescrit dans le cycle de workflow et les informations relatives à l'exécution de chaque phase.

L'écran 2 illustre l'interface graphique appelée **UserWindow** qui est propre à chaque utilisateur. Cette interface contient la liste des patients admis dans le service et la liste des «post-it» de coordination et/ou libres qui sont destinés à l'utilisateur connecté. L'affichage des «post-it» dépend du patient sélectionné. L'utilisateur peut accéder à l'interface **WorkflowItem** en double-cliquant sur le nom d'un patient donné ou sur un «post-it» de coordination (ce type de «post-it» contient une référence à un patient donné).

modèle contient les structures de données et les méthodes adéquates et nécessaires pour l'exécution d'une action donnée. Le résultat de chaque opération est renvoyé par le modèle vers la vue pour l'affichage.

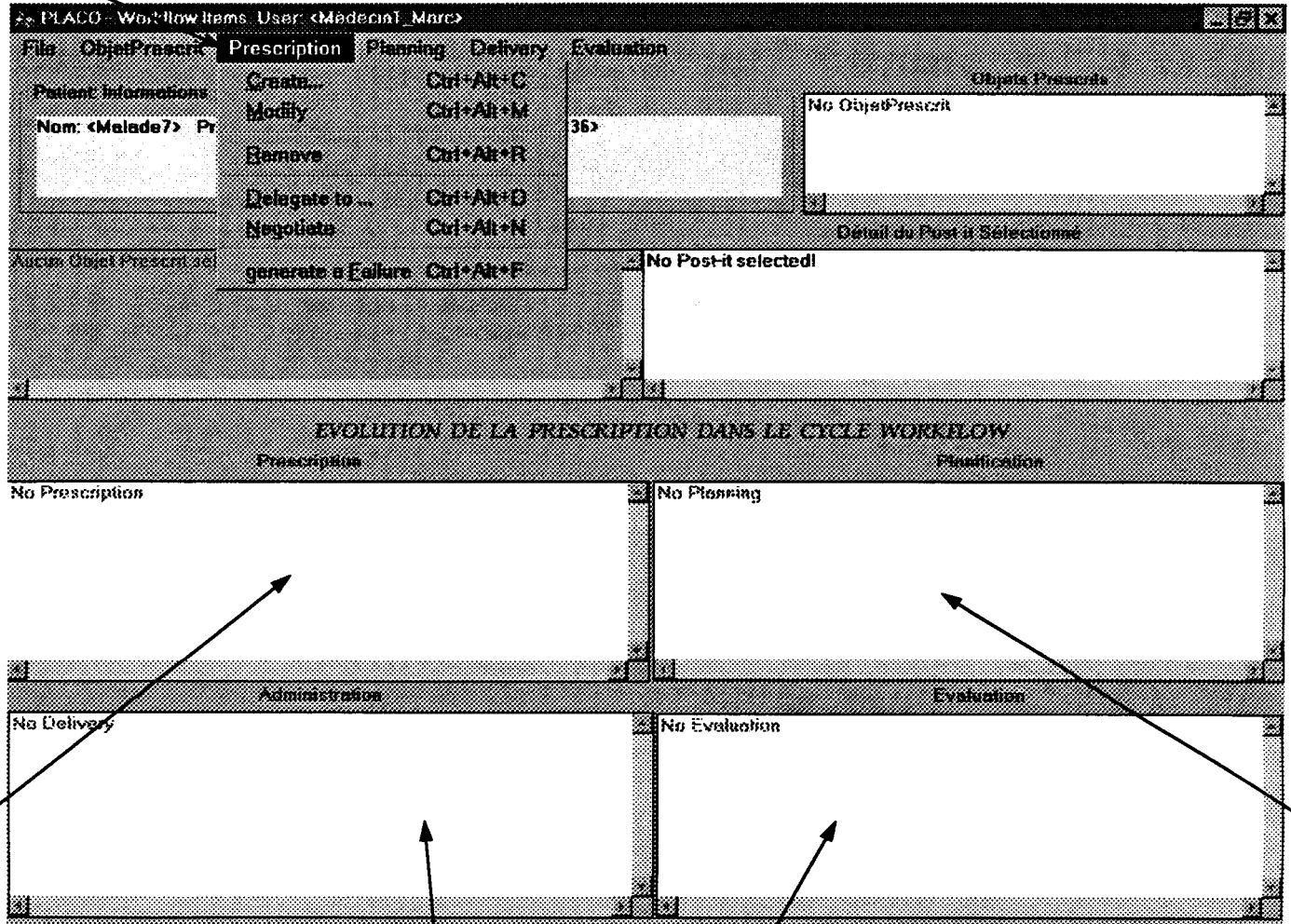
Afin d'illustrer cette approche, nous nous proposons de montrer les copies d'écran de deux interfaces graphiques que nous avons réalisées en respectant le modèle MVC. Toutefois, pour plus de détail sur le contenu des interfaces et les fonctionnalités qu'elles offrent, nous renvoyons le lecteur à l'annexe B.

L'écran 1 illustre l'interface graphique appelée **WorkflowItem** qui est propre à un patient. Elle contient la liste des objets prescrits, la liste des «post-it» de workflow relatifs à l'état de chaque objet prescrit dans le cycle de workflow et les informations relatives à l'exécution de chaque phase.

L'écran 2 illustre l'interface graphique appelée **UserWindow** qui est propre à chaque utilisateur. Cette interface contient la liste des patients admis dans le service et la liste des «post-it» de coordination et/ou libres qui sont destinés à l'utilisateur connecté. L'affichage des «post-it» dépend du patient sélectionné. L'utilisateur peut accéder à l'interface **WorkflowItem** en double-cliquant sur le nom d'un patient donné ou sur un «post-it» de coordination (ce type de «post-it» contient une référence à un patient donné).

Les différentes actions réalisables sur les objets prescrits :
création, modification, suppression, délégation de tâche, clarification de tâche et génération d'arrêt.

Les menus déroulants concernent la prescription, la planification, l'administration et l'évaluation d'un objet prescrit.



Affiche les informations concernant la prescription d'un médicament

Affiche les informations concernant l'administration d'un médicament

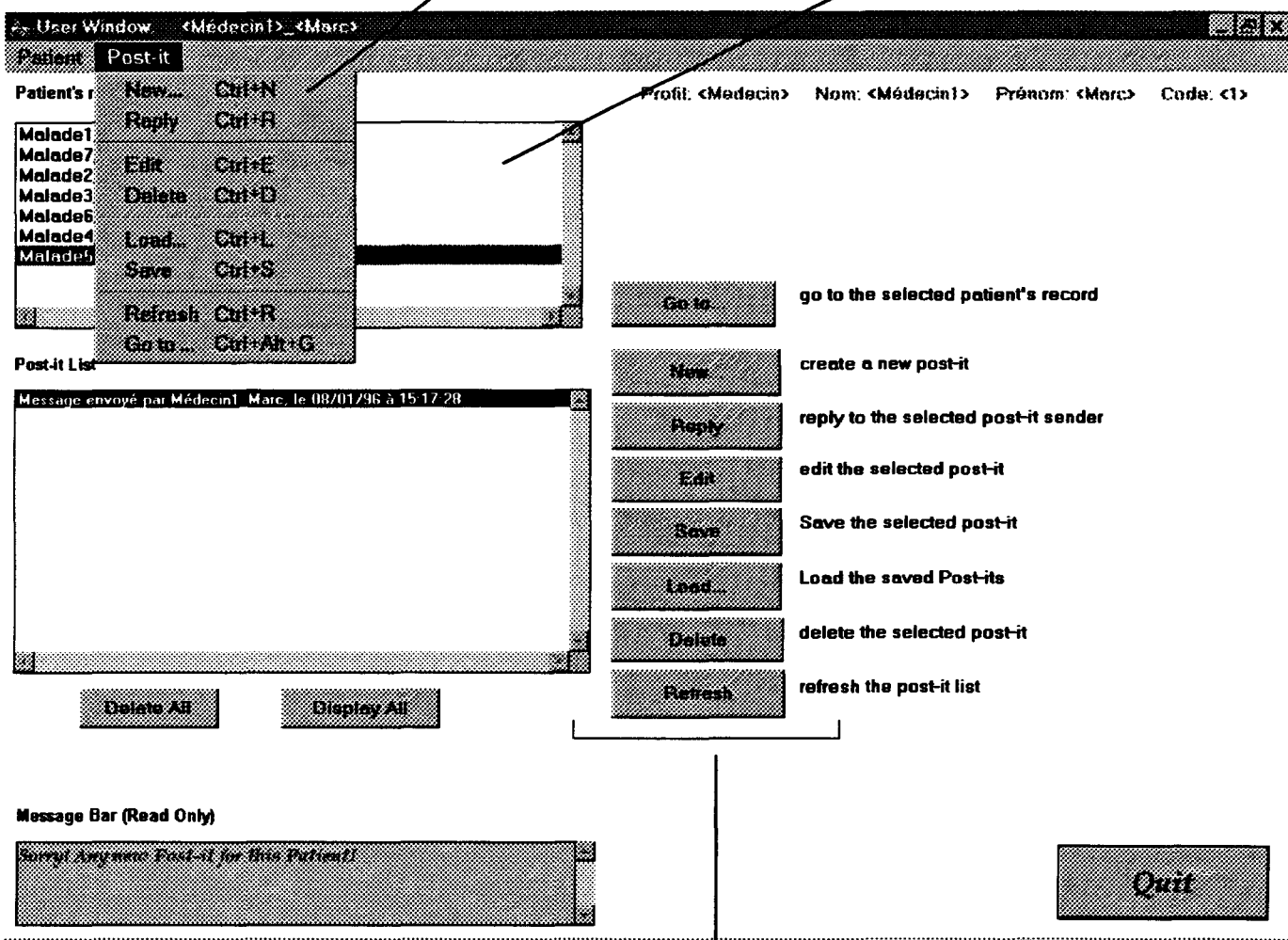
Affiche les informations concernant la planification d'un médicament

Affiche les informations concernant l'évaluation d'un médicament

Ecran 1 : La fenêtre WorkflowItems qui permet à l'utilisateur « Médecin1 Marc » de manipuler les objets prescrits et de voir leur évolution au fil du temps.

Menu déroulant offrant la possibilité d'effectuer un certain nombre d'opération sur les « post-it ».

Liste des patients admis dans le service



Ensemble de fonctionnalités offertes depuis la fenêtre UserWindow: accéder à un patient, créer un nouveau « post-it », répondre à un « post-it », sauvegarder un « post-it », recharger les « post-it » stockés, etc.

Ecran 2 : Fenêtre UserWindow qui permet à l'utilisateur « Médecin1 Marc » de consulter les « post-it » qu'il a reçus et d'accéder aux dossiers des patients qui contiennent la liste des objets prescrits et leur évolution dans le cycle de workflow. L'accès aux dossiers des patients se fait soit par voie de « post-it » (en double-cliquant sur le « post-it » adéquat), soit par voie du patient (en double-cliquant sur le nom du patient en question).

7.3 Evaluation du Prototype PLACO

L'évaluation médicale du projet PLACO a été réalisée de deux manières. D'une part, nous devons envisager une évaluation conceptuelle des solutions proposées en s'assurant de leur adaptation à la réalité médicale choisie. Ceci consiste en la validation, par des médecins ou des infirmier(e)s, des modèles, de l'architecture et de l'ergonomie des solutions envisagées. D'autre part, il faut une évaluation clinique sur site, ce qui nécessite l'installation, sur le site d'évaluation, de plusieurs postes (au moins trois) avec des interfaces finalisées.

Il semble intéressant qu'un outil supportant le travail coopératif en USI prenne en compte l'ensemble des données qui concerne le patient. Par conséquent, il doit non seulement gérer les prescriptions, mais également les paramètres de surveillance, les examens complémentaires et les actes médico-techniques. L'outil coopératif devrait être capable de supporter l'ensemble du travail de l'équipe de soin.

Le but du projet PLACO est de valider un modèle et une architecture, et non de développer un produit fini. Dans le chapitre 4, nous avons montré que le domaine sensible en USI est celui de la prescription (voir aussi les travaux rapportés dans les «deliverables» [AIM-TANIT93] [AIM-TANIT94] et [Yous95d]). C'est pourquoi l'application du modèle proposé a été restreinte au domaine de la planification coopérative.

7.3.1 Le modèle PLACO

Le modèle de Workflow que nous avons proposé semble être bien adapté à la description de l'activité de soins [AIM-TANIT94]. Celle-ci est en effet régie par les prescriptions médicales. Chaque prescription est réalisée en prenant en compte les données actuelles et passées concernant le patient (évaluation des phases précédentes). La transition avec la phase d'accord permet de vérifier que l'objet de la prescription est bien réalisable (disponibilité du matériel et/ou du personnel, planification, etc.).

Le modèle de workflow, par son fonctionnement cyclique et par l'imbrication de sous-cycles de **délégation** et de **clarification**, permet de rendre compte du déroulement complet des soins. Les «breakdown» sont ainsi traités par la génération de sous-tâches. Leur mise en oeuvre la plus simple consiste en un processus de génération de «post-it» à destination

du prescripteur ou de l'acteur concerné. La nécessaire flexibilité de la répartition des tâches au sein des équipes amène une précision. La notion d'acteur dans le cycle de workflow ne fait pas référence à un individu nommément désigné, mais à la personne qui remplit la fonction pour une tâche donnée à un instant donné.

A défaut de supprimer totalement les «breakdown», la mise en oeuvre du modèle de workflow permet de les traiter dès leur apparition. Certains «breakdown», comme une erreur d'administration de médicament, ne peuvent pas être supprimés par le système. Ils doivent donc être traités par un opérateur.

7.3.2 Evaluation des fonctions du prototype PLACO

Cette évaluation a eu pour objectif d'analyser les fonctions offertes par PLACO, de les commenter et de suggérer des adaptations propres au milieu médical. Cette évaluation a été réalisée en laboratoire et non en USI, par un médecin de travail, spécialisé en ergonomie du travail, et attaché au département d'information médicale pour lequel il a mis au point les méthodologies d'évaluation technique et ergonomique. Par ailleurs, le docteur B. Bossard [boss92] a été responsable de l'évaluation technique des projets européens dans le programme ISAR dont le CHRU de Lille est contractant principal [ISAR94a].

7.3.2.1 Déroulement de l'évaluation

Les grandes fonctionnalités de PLACO ont été évaluées en laboratoire en soulignant les points positifs et les améliorations à apporter, que ce soit au niveau de l'Interface Homme-Machine (IHM) ou au niveau des fonctionnalités de l'application elle-même.

Il s'agit là d'une évaluation des **fonctions** offertes par le prototype et non d'une évaluation technique exhaustive. Cette évaluation consiste à:

- observer tout ce que peut apporter PLACO de par ses fonctions au niveau de l'activité d'un médecin (ou d'une infirmière), et aussi au niveau de l'activité globale d'un service médical,
- commenter ces observations,
- émettre certaines propositions.

7.3.2.2 Résultats de l'évaluation

Les résultats que nous présenterons couvrent plusieurs aspects du prototype. D'abord, nous parlerons de la coordination entre acteurs, ensuite du prototype en tant qu'outil de communication, et enfin de la gestion et du suivi des patients.

A. La coordination entre les acteurs de soins

1. La répartition des tâches entre acteurs dans le cycle workflow principal

- **Description:** pour effectuer les différentes phases du workflow principal, aucune répartition des tâches entre acteurs n'est pas imposée par le prototype. Il laisse donc les acteurs (médecins et infirmières) libres de s'organiser pour effectuer le cycle complet, à savoir *qui fera telle action dans le cycle*.
- **Commentaires:** ceci représente un atout au niveau de l'intégration du prototype dans les services médicaux. Mais cette grande souplesse de l'application au niveau de la répartition des tâches, si elle est considérée comme un aspect positif, peut être source de quelques dysfonctionnements si une organisation minimale n'est pas instaurée, à savoir *qui fait quoi*. Par exemple, qui sera **responsable** des actions de planification, administration, etc?
- **Proposition:** les utilisateurs doivent être mis au courant de cette liberté d'action. Ils doivent être sensibilisés au fait qu'ils seront leur propre responsable pour l'organisation des activités de soins. On doit aussi pouvoir imposer des contraintes pour affecter certaines actions à des acteurs définis.

2. Les fonctionnalités de délégation et de négociation

- **Description:** elles permettent un transfert des tâches vers d'autres acteurs potentiels. Aucune contrainte n'existe (une demande de délégation peut être envoyée n'importe quand, à n'importe qui et pour n'importe quelle action.). Elles permettent aux acteurs de gérer avec souplesse les contraintes liées à leur activité.
- **Commentaires:** l'absence de contrainte au niveau de l'application peut être à l'origine de certains dysfonctionnements (si les médecins d'un service s'envoient mutuellement une demande de délégation, que faire dans ce cas?).
- **Propositions:** une visualisation de l'ensemble des demandes de délégation et/ou de négociation pour un acteur donné, doit être présente afin que ce dernier puisse avoir un

rappel global de ce qu'il a fait et de ce qu'il a prévu de faire, ou de ce que d'autres acteurs lui demandent. Il serait également intéressant de laisser au système le soin de gérer dynamiquement les contraintes de disponibilité des acteurs.

3. L'accès indifférencié de PLACO par les médecins et les infirmières

- **Description:** pour les médecins et les infirmières ayant accès à PLACO, l'application représente un nouveau référentiel d'échange d'informations entre des acteurs possédant des rôles différents. En effet, PLACO regroupe sur un même support des informations qui existent habituellement sur des supports différents, avec des formes de présentation différentes. Par exemple, l'administration d'un médicament et ses éventuelles conséquences figurent habituellement au niveau d'un dossier patient infirmier, alors que la planification figure au niveau d'un dossier thérapeutique.
- **Commentaires:** ce partage d'informations (apport essentiel de PLACO) ne sera pas sans conséquence au niveau:
 - de la clarté de l'information échangée,
 - du mode de travail des médecins et des infirmières,
 - de la qualité des informations échangées.

4. La gestion des «breakdown»

- **Description:** l'application donne la possibilité à chaque acteur de soins de noter tous les problèmes liés à l'administration des prescriptions.
- **Commentaires:** la possibilité d'intégrer ces informations au niveau du dossier médical du patient ne pourra être que bénéfique pour ce même patient. Nous abordons ici la problématique de la création d'un *système d'information* lié à la prescription pouvant s'intégrer dans une démarche de *qualité des soins*.
- **Proposition:** si l'un des objectifs de PLACO est d'optimiser la coordination entre les différents acteurs de soins, cette possibilité ne doit pas être négligée dans la conception des systèmes d'information. Une fonctionnalité, permettant de regrouper tous les «breakdown» et les problèmes liés aux prescriptions, serait intéressante pour assurer cette démarche de qualité de soins.

5. Visibilité des «breakdown»

- **Description:** en cas de «breakdown», il existe un message spécifique qui apparaît.
- **Commentaires:** un des grands aspects positifs de l'application est la gestion des «breakdown». Les médecins y attacheront un grand intérêt dans le cadre d'une optimisation de la qualité des soins, de la pharmacovigilance, du suivi thérapeutique, etc.
- **Proposition:** les informations qui font suite à un «breakdown» doivent revêtir certaines caractéristiques. Elles doivent:
 1. être facilement «différenciables» par rapport aux autres types d'informations (en particulier les «post-it»),
 2. être d'une grande accessibilité pour l'utilisateur.

B. PLACO et les outils de communication

1. Les différents types de «post-it»

- **Description:** les «post-it», qui offrent la possibilité d'échanger directement des informations entre utilisateurs, sont indépendants du cycle workflow principal ou secondaire («post-it» libres).
- **Commentaires:** cette fonctionnalité semble être une option intéressante car elle permettra aux acteurs d'échanger des compléments d'information. Cependant, le prototype, dans sa version actuelle, ne permet pas une différenciation nette entre les «post-it» de coordination, les «post-it» de workflow et ceux dits libres.
- **Proposition:** il faudra mieux différencier, vis-à-vis des utilisateurs, les trois types de «post-it», ainsi que les «post-it» du même type, en particulier ceux liés aux objets prescrits dans les workflows. Par exemple, un mode de présentation différent, une dénomination différente, etc, seraient un atout considérable pour l'application.

2. Gestion des «post-it»

- **Description:** le gestionnaire de «post-it» remplit la fonction de boîte aux lettres (i.e. boîte de «post-it»), qui sert à recevoir des «post-it» et à les stocker à l'initiative de l'utilisateur.
- **Commentaires:** ceci se révèle intéressant car il:
 1. permet de «vider» la boîte aux lettres à tout moment sans perdre trace des anciens «post-it». Tout nouveau message sera plus facilement identifiable.
 2. permet de garder une trace de certains «post-it» que l'utilisateur considère importants.
- **Proposition:** au niveau de l'interface, la distinction et les relations entre les «post-it» reçus et ceux stockés, doivent être plus évidentes pour l'utilisateur.

3. Stockage des «post-it»

- **Description:** la boîte aux lettres ne permet pas de classer les «post-it» par ordre d'urgence ou de priorité.
- **Commentaires:** les «post-it» qui semblent moins importants pourraient être traités ultérieurement. La fonction de stockage devrait permettre à l'utilisateur d'organiser son travail en mettant en attente les «post-it» non urgents, par exemple.
- **Proposition:** pour que l'acteur puisse apprécier ses possibilités de planification, il serait intéressant qu'avec le type du «post-it», apparaisse sa date de création ainsi que sa durée de validité, au niveau de la boîte aux lettres.

Mais attention, s'agissant d'une boîte de stockage personnelle, non mise à jour de manière automatique, certains «post-it» peuvent devenir obsolètes. Par exemple, un «post-it» de prescription peut être mis en attente par un médecin. Pendant ce temps, un autre médecin réalise la prescription, ce qui génère un «post-it» de planification pour le même patient. Dans ce cas, le premier «post-it» devient obsolète.

4. Destruction des «post-it»

- **Description:** l'application donne la possibilité au médecin de supprimer les «post-it» reçus sans aucune contrainte pour lui.
- **Commentaires:** cette souplesse met en avant les problèmes de coordination du travail et de transfert de responsabilités.

Ainsi, toute trace d'un «post-it» (de prescription, par exemple) généré par un médecin, peut être perdue si tous les médecins décident de détruire ce «post-it». Ce dernier peut être retrouvé, sous une autre forme, au niveau du dossier patient, mais encore faut-il y penser!

C. le suivi thérapeutique

1. Suivi du workflow pour un patient

Il faut souligner d'emblée quelques aspects positifs de l'interface homme-machine du prototype, aspects positifs qui, à notre avis, doivent être considérés comme fils conducteurs pour la réalisation future d'une application.

- **Description:** le prototype permet la visualisation de la totalité du cycle workflow pour un objet prescrit sur un seul écran.
- **Commentaires:** cet aspect permet une rapidité et une facilité d'accès à l'information, surtout dans le cadre de la création d'un nouveau référentiel d'informations (notion que nous avons introduite précédemment). Par contre, la visualisation de tous les objets prescrits est une tâche qui se révèle extrêmement lourde pour le médecin, car il doit réitérer l'opération de visualisation pour chaque objet (chaque médicament).
- **Proposition:** une amélioration du prototype sera donc de fournir une fonctionnalité permettant pour chaque patient une visualisation simultanée de tous les objets prescrits afin de voir rapidement:
 - où en sont-ils au niveau de leur cycle?
 - quels sont les problèmes rencontrés?

De même, il serait intéressant, notamment pour les infirmières, de disposer d'une planification globale des prescriptions qui permettrait de mieux voir les incompatibilités entre les différents médicaments prescrits.

2. Actions possibles sur les objets

- **Description:** le prototype offre une homogénéité et un faible nombre d'actions possibles pour chaque phase. Six actions sont proposées: «*create*», «*modify*», «*remove*», «*negociate*», «*delegate*» et «*generate a failure*».

- **Commentaires:** cette caractéristique lui confère une facilité d'utilisation, mais aussi une rapidité de maîtrise de l'outil. Ces propriétés sont très intéressantes pour des utilisateurs occasionnels.
- **Proposition:** une amélioration possible vis-à-vis des utilisateurs experts sera de proposer une sorte de «boîte à outils», qui soit disponible à tout moment pour chaque phase du cycle, et qui comprenne ces six fonctionnalités.

3. Suivi du Workflow pour plusieurs patients

- **Description:** grâce à l'«iconisation» du dossier patient, le médecin ou l'infirmière, a la possibilité de pouvoir gérer plusieurs dossiers patient à sa convenance, et donc de visualiser en un minimum de temps l'évolution de la prescription pour tout patient.
- **Commentaires:** on peut attendre de cette possibilité de grandes améliorations au niveau:
 1. du temps d'accès à l'information désirée,
 2. de l'adaptabilité de l'application en fonction des situations de travail. Chaque situation engendrera de la part de l'acteur une organisation et une planification différentes.

Ainsi, par exemple, l'«iconisation» des patients soumis à une thérapie particulière permettra à l'acteur responsable de suivre de façon optimale l'évolution.

- **Proposition:** dans le cadre de la planification coopérative, de nouvelles fonctionnalités semblent intéressantes à fournir:
 1. L'«iconisation» du dossier patient avec l'apparition du nom du patient et non seulement celui de l'utilisateur.
 2. La possibilité de réaliser des groupements de dossiers médicaux.

En définitive, pour éliminer les difficultés liées à l'introduction de l'informatique dans les unités de soins, indépendamment du système proposé, il faut disposer d'une interface finalisée dont l'ergonomie influera le moins possible sur l'évaluation de l'architecture proposée.

7.4 Perspectives

Dans ce paragraphe, nous présenterons, en terme de perspectives, les travaux à court-terme qui restent à effectuer dans le cadre du projet PLACO. Il s'agit de la distribution de l'application et de l'amélioration des interfaces graphiques. Ces améliorations sont déterminantes quant à une réelle évaluation sur site du prototype.

7.4.1 La distribution de l'application

Le prototype que nous avons réalisé est une application qui fonctionne en mono-utilisateur. L'objectif que nous avons visé, dans un premier temps, est d'élaborer un modèle stable qui régit les interactions humaines dans une unité de travail. Cependant, pour une évaluation plus concluante qui nous permettrait d'améliorer efficacement les fonctionnalités de notre prototype, nous nous devons de réaliser une deuxième version qui tienne compte de cette dimension qu'est la répartition.

L'étape de spécification et de définition du graphe d'héritage complet des différents objets manipulés, dans la solution PLACO, est déjà effectuée (voir le paragraphe 7.1). Il nous appartient donc de réutiliser, dans l'étape suivante consistant en la distribution de l'application, la définition et la description de ces objets, ainsi qu'une partie du code (si possible) de la première version. Dans cette optique, nous suggérons l'utilisation du langage DSTTM (version de Smalltalk distribué de HP), qui fonctionne sous UNIX et qui utilise la norme CORBA, pour implanter l'application distribuée de la solution PLACO.

7.4.2 Les interfaces homme-machine

Les nouvelles possibilités offertes par les Groupware doivent être intégrées dans le travail des acteurs accomplissant leur tâches. Dans le cas de la bureautique, cette nécessité est évidente car l'ordinateur représente l'élément clé du travail de la secrétaire. En revanche, dans les USI, l'ordinateur n'est pas encore considéré comme indispensable pour la gestion de l'information [AIM-TANIT94]. C'est pourquoi, il est important d'insister sur les interfaces qui faciliteront considérablement la manipulation de l'information et l'amélioration du travail.

Pour l'amélioration des interfaces graphiques du prototype, nous suggérons d'utiliser la métaphore du «post-it» pour réaliser ces interfaces. D'après l'évaluation technique

détaillée précédemment, il en découle clairement que les acteurs de soins exigent de manipuler le minimum de fenêtres avec le maximum d'information. Or, la première propriété va, parfois, à l'encontre de l'autre.

Il faut également prendre en compte le fait que les acteurs remplissent des rôles différents, et donc n'ont pas forcément la même vue sur la même information. Une information peut être vue de différentes façons, par exemple, un médecin classe les médicaments selon leur principe actif. Par contre, l'infirmière les classe selon leur voie d'administration (PO, SAP, IV, SC). Il s'ensuit donc que la conception des interfaces doit prendre en compte cette différenciation et filtrer l'information selon le rôle de l'utilisateur à un instant donné.

Conclusion et Perspectives

Le travail est un phénomène impliquant plusieurs formes d'interactions sociales [Schm91]. L'étude des systèmes dédiés à supporter un travail commun à plusieurs personnes a donné naissance à une nouvelle discipline scientifique appelée le CSCW: *Computer-Supported Cooperative Work*, acronyme dont la traduction en français donne le *Travail Coopératif Assisté par Ordinateur* ou encore, d'après l'AFCEC, *Activités Multi-Participantes Informatisées*.

Les systèmes CSCW qui ont pour objectif de soutenir la nature sociale d'une organisation introduisent un changement technologique, parfois abrupt, qui modifie l'organisation sociale du travail du groupe. Or, ce dernier manifeste un refus naturel afin de préserver sa «*culture sociale*», et l'évaluation de tels systèmes dans des cas réels s'avère très difficile. Les résultats obtenus sont presque hypothétiques et rarement concluants à cause de la variabilité de la dynamique des groupes [Grud88].

En réalité, ni le travail de groupe, ni les technologies associées ne sont nouvelles. Cependant, le CSCW a permis de reconsidérer plusieurs aspects qui contribuent à la conception des technologies destinées à supporter le travail de groupe. En particulier, il a soulevé les questions qui traitent du support de travail commun au groupe [McCa91], et de la coordination de leurs activités: *qui peut faire quoi, où et quand?* Ces deux aspects du travail de groupe relèvent de la «*conscience collaborative*» [Proc94].

Les groupes d'individus peuvent réaliser un travail de routine sans faire appel à la notion de «*conscience collaborative*». En revanche, lorsque les tâches de groupe s'inscrivent

dans un contexte de résolution intelligente de problèmes, le besoin de partager l'information et de coordonner les activités de façon interactive devient primordial [Proc94].

Dans les applications multi-utilisateurs conventionnelles, le principe de la conception consiste à dissimuler le fait que les ressources, qu'elles soient de type système ou d'information, sont partagées entre des dizaines, des centaines, voire des milliers d'utilisateurs. Par exemple, une base de données offre le support du travail commun, mais elle ignore complètement le problème de coordination en rendant anonymes les actions des utilisateurs. Elle assume uniquement la responsabilité de leur coordination à fin grain (niveau système). Si la coordination de leurs activités est nécessaire à d'autres niveaux, ils doivent assumer eux-mêmes cette fonction en utilisant tous les mécanismes et les ressources disponibles et extérieures au système.

PLACO (PLAnification COopérative) est un système informatique destiné à supporter la planification coopérative dans les unités de travail, et en particulier les unités médicales (anesthésie-réanimation). Notre modèle doit tenir compte de la dimension humaine et décrire les interactions et échanges conversationnels entre les acteurs d'une unité de travail.

L'introduction de nouvelles technologies pour la coordination des activités et la gestion des conflits dans les unités de travail requiert la prise en considération de plusieurs facteurs de nature distincte tant au point de vue humain que matériel ou organisationnel. La complexité de ce domaine nous a d'abord amenés à élaborer un modèle coopératif, dans lequel nous avons défini les relations de dépendance entre les tâches requises et les rôles impliqués, en vue de prévenir à temps les «breakdown» et d'éviter les redondances dans le processus de travail.

L'objectif à terme, visé par le projet PLACO, est de développer un outil Groupware distribué permettant de supporter la coopération entre individus, et de résoudre les problèmes de coordination de leurs activités au sein d'une organisation de travail.

Au sein des organisations à faible couplage (entre les acteurs et les tâches), telles que les USI, les participants sont très mobiles. Ils sont «distribués» dans l'espace et dans le temps.

Afin de garantir une cohésion de l'ensemble des processus de travail, ils doivent coordonner efficacement leurs activités.

L'investigation des théories et modèles de coordination nous a donc amené à privilégier l'approche orientée conversation de WINOGRAD et FLORES [Wino87b]. Cette approche, qui se base sur la gestion du flux de travail (workflow), met particulièrement l'accent sur les interactions humaines et le transfert de responsabilités entre individus. Elle présente l'avantage de décomposer naturellement les activités sous forme de scénarii coopératifs récurrents et de spécifier les acteurs impliqués. Elle offre également aux acteurs la possibilité de savoir ce qui s'est passé et ce qui va se passer dans chaque processus de travail. Ainsi, tout acteur peut disposer d'une vue globale de l'ensemble des tâches effectuées et de celles qui restent à effectuer. Par conséquent, ils peuvent éviter et prévenir à temps les incidents de fonctionnement.

Nous avons donc défini un modèle de Workflow, dérivé de celui de WINOGRAD et FLORES, qui prend en charge les interactions des utilisateurs au sein d'une unité de travail, et nous l'avons généralisé pour qu'il soit applicable à toute organisation où la coordination est de règle. Or, à cause du caractère dynamique inhérent à ces interactions [Proc94], nous avons introduit dans ce modèle la notion de *lien dynamique* entre l'acteur et les tâches dont il est responsable. Ce type de lien permet l'allocation dynamique des tâches, dans le temps, aux différents acteurs qui coopèrent au sein d'une organisation. Nous avons ainsi mis l'accent sur le *rôle* qu'un acteur doit remplir, au lieu de le mettre sur l'individu lui-même.

Par ailleurs, nous avons étendu notre modèle afin de prendre en compte d'autres formes d'interaction nécessaires à l'accomplissement efficace et efficient des actions requises. Il s'agit de la *délégation* et de la *clarification* dynamiques des tâches auxquelles un acteur peut avoir recours à tout moment. Ces deux formes d'interaction permettent d'éviter la suspension d'un processus de travail chaque fois qu'un acteur est indisponible pour une raison quelconque, ou chaque fois que manque une information indispensable à la réalisation d'une tâche.

En outre, notre modèle offre aux acteurs la possibilité de générer explicitement un *arrêt* du processus de travail en cours. Cette possibilité de «*génération d'échec*» permet de forcer l'exécution de la phase d'évaluation du travail mis en échec, afin d'en déceler les causes.

Le mécanisme de coordination que nous avons mis en place dans le modèle se base sur l'utilisation des messages semi-structurés. Ce type de message se révèle, à la fois, naturel, simple et efficace en matière de gestion et d'implantation physique [Malo86a]. De plus, ce mécanisme favorise la *communication informelle* entre acteurs comme avec les messageries classiques.

L'architecture proposée, pour le modèle PLACO, est une architecture multi-couche, coopérative et distribuée. Elle permet deux types de contrôle: le contrôle de cohérence des objets manipulés et celui des interactions entre utilisateurs. Bâtie autour d'un espace d'information partagé, elle permet aux acteurs d'interagir et de manipuler des objets d'information représentant le domaine. Chaque objet d'information s'inscrit dans un cycle appelé **cycle de workflow**, allant de la phase de prescription jusqu'à la phase d'évaluation, via la planification et la réalisation.

Le prototype que nous avons développé, selon une approche incrémentale, s'applique à la **PLAnification COopérative** (d'où le nom PLACO) au sein des unités de soins intensifs. Il permet aux acteurs de soins, médecins et infirmières, de réaliser efficacement des prescriptions médicales, et de les administrer aux patients. Ce type d'activité requiert une coordination efficace et incessante, afin d'éviter tout incident susceptible de perturber le fonctionnement de l'unité.

Le contrôle, dans le prototype, se base sur le concept d'agent. Il représente une entité autonome et indépendante du domaine d'information. Nous avons conçus trois types d'agent: agent d'interface, agent de workflow et agent de messages.

La prise en charge de la coordination est réalisée par l'agent des messages. Dans notre approche, nous désignons ces messages par des «post-it». Cette métaphore présente un grand intérêt pour la conception des interfaces-utilisateurs. Les «post-it», du fait que leur contenu informationnel dépend de l'état de workflow de l'information, ont la faculté d'être attachés, soit aux objets d'information, soit aux acteurs. Ce point innovant permet une compréhension aisée du rôle des «post-it» dans la coordination des activités.

La séparation entre le contrôle et le domaine présente, pour notre prototype, l'avantage d'avoir un outil *évolutif et réutilisable* à d'autres types d'application tels que la gestion des

projets, l'ordonnement des activités au sein d'une structure de recherche distribuée («éclatée») telle que Ganymede, etc.

Les bénéfices potentiels que les utilisateurs finaux sont en droit d'attendre des outils coopératifs, tel que le système PLACO, consistent en:

- une meilleure efficacité des processus de prise de décision,
- une optimisation des ressources,
- une résolution rapide des situations de crise,
- une diminution des «breakdown».

Pour les unités médicales, la prescription des actes thérapeutiques apparaît comme un domaine d'application privilégié des méthodes et outils du travail coopératif, parce qu'elle est sujette à de fréquentes modifications et adaptations et qu'elle implique l'ensemble des acteurs de l'unité de soins.

En définitive, en l'état actuel des travaux de recherche menés dans le projet PLACO, nous pouvons exhiber un modèle coopératif régissant les interactions humaines au sein des unités de travail. Ce modèle est innovant en ceci qu'il intègre:

- une dimension cognitive issues des travaux sur la psychologie cognitive,
- une dimension informatique consistant en la gestion de workflow issue des travaux de recherche de WINOGRAD et FLORES [Wino87b],
- un mécanisme de coordination des tâches inspiré des travaux de recherche de MALONE [Malo90].

Perspectives

L'approche de PLACO couvre les aspects théoriques, techniques et pratiques du projet, depuis la phase d'investigation jusqu'à la réalisation d'un prototype et à son évaluation. Cependant, PLACO reste un projet de recherche destiné à explorer l'importance du travail coopératif et son impact potentiel pour la définition et la réalisation de logiciels dédiés à des unités de travail où la dimension de «*collaboration entre acteurs*» est particulièrement présente.

Le prototype que nous avons réalisé permet, au stade de son développement, de réaliser des simulations de cas concrets. Il nous reste à faire évoluer, notamment, les aspects des interfaces graphiques et de la distribution, pour obtenir un prototype stable qui peut être installé dans l'environnement réel d'une unité d'anesthésie et de réanimation, à des fins de validation et d'expérimentation.

D'autres travaux devront s'intéresser à la représentation de connaissances dans un environnement distribué tel que les USI, à l'étude des problèmes liés à l'automatisation de l'allocation des tâches aux différents acteurs, à la modélisation des contraintes temporelles et spatiales inhérentes aux acteurs de soins. Un troisième aspect, non moins important, concernera l'étude et l'analyse des problèmes d'intégration d'une telle application dans un système d'information hospitalier tel que **OTALIA** (système d'information hospitalier du CHRU de Lille).

Références Bibliographiques

-
- [ATI89] ATI: Action Technologies Inc.
«*The Coordinator II*»
Action Technologies Inc., 1989.
- [ATI94] ATI: Action Technologies Inc.
«*Action Workflow Analyst's User's Guide*».
Action Technologies Inc., 1994.
- [AIM-TANIT92] AIM-TANIT: R. Beuscart, V. Carre, F. Westrelin, S. Dalmas et al.
«*Interim Report on Cooperative Work in CEE: Methodology*».
AIM-TANIT (A2036 Project), COOP Workpackage. Deliverable #D2, July 1992.
- [AIM-TANIT93] R. Beuscart, M. Demeester, E. Dufresne, F. Yousfi et al.
«*Methodologies for Development of Cooperative Work*».
AIM-TANIT (A2036 Project), COOP Workpackage. Deliverable #D15, July 1993.
- [AIM-TANIT94] R. Beuscart, F. Yousfi, E. Dufresne, J-M. Geib et al.
«*Proposal for Cooperative Work Support in Critical Care Departments*».
AIM-TANIT (A2036 Project), COOP Workpackage. Deliverable #D27, July 1994.

- [Appl86] L. M. Applegate, B. R. Konsynski and J. F. Nunamaker.
«*A Group Decision Support System for Idea Generation and Issue Analysis in Organization Planning*».
In Proceedings of the First Conference on Computer-Supported Cooperative Work, Austin, Texas, December 3-5, ACM, New York, 1986, pp 16-34.
- [Arms90] M. Armstrong.
«*Management Processes and Functions*».
In M. Armstrong, D. Farnham (eds): Management Studies Series, 1990.
- [Aust62] J. Austin.
«*How To Do Things With Words*».
Cambridge, Massachusetts: Harvard University Press, 1962.
- [Bann89] L. J. Bannon and K. Schmidt.
«*CSCW: Four Characters in Search of a Context*».
In Proceedings of the European Community Conference on CSCW (EC-CSCW), London, September 1989, pp. 358-372.
- [Beus94] R. Beuscart, F. Yousfi, E. Dusfresne et A. Derycke.
«*Travail Coopératif et Groupware*»
J. Demongeot, P. Le Beux, et G. Weil (rédacteurs), Informatique et Santé: Informatisation de l'Unité de Soins du Futur, Vol. 7, 1994, pp. 195-209, Springer-Verlag, Paris.
- [Blaz89] R. Blazer.
«*Process Programming: Passing into a New Phase*».
In Proceedings of the Fourth International Software Process Workshop, Devon, UK, May 11-13, ACM SIGSOFT 14, June 4, 1989.
- [Blum90] B. I. Blum and K. Duncan.
«*A History of Medical Informatics*».
Bruce I. Blum and Karen Duncan (eds). ACM Press, New York, 1990.

-
- [Booc94a] G. Booch.
«*Object-Oriented Design with Applications*».
Benjamin/Cummings, New York, 1994.
- [Booc94b] G. Booch.
«*Analyse et Conception Orientées Objets*».
Addison-Wesley Eds. Paris, 1994.
- [Boss92] B. Bossard.
«*Méthodologies de Développement d'une Application Multimedia à Visée Coopérative en Milieu Médical*».
Thèse de Médecine, Université de Lille II, 1992.
- [Bowe95] C. L. Bowes and J. Holland.
«*TANIT: Telematics for Anaesthesia and Intensive Therapy*».
Health in the New Communications Age: Health Care Telematics for the 21st Century. M. F. Laires, M. J. Ladeira and J. P. Christensen (eds), IOS Press, Amsterdam, 1995.
- [Boy88] G. Boy.
«*Assistance à l'Opérateur: Une Approche de l'Intelligence Artificielle*».
Teckenea, 1988.
- [Bric94] N. Bricon-Souf.
«*ARCICE: Architecture Réflexive d'un Compèrobot Interactif, Compréhension des Enoncés*».
Thèse d'Informatique, LAIAC, Université des Sciences de Caen, mars 1994.
- [Carr89] B. Carré.
«*Méthodologie Orientée Objets pour la Représentation des Connaissances, Concepts et Point de Vue, de Représentation Multiple et Evolutive d'Objet*».
Thèse d'Informatique, LIFL, Université des Sciences et Technologies de Lille, 1989.

- [Cent93a] Katy Center and Suzanne Henry.
«*A New Paradigm for Business Processes*».
In Proceedings of the Workflow Conference on Business Process Technology, San-Jose, California, August 1993. pp 167-177. Morgan Kaufmann Publishers Inc.
- [Cent93b] Katy Center and Cynthia Scoggins.
«*Cultural Transformation through the Workflow Paradigm*».
In Proceedings of the Workflow Conference on Business Process Technology, San-Jose, California, August 1993. pp 179-185. Morgan Kaufmann Publishers Inc.
- [Conk87] J. Conklin.
«*gIBIS: A Hypertext Tool for Team Design Deliberation*».
In Proceedings of Hypertext'87, November 1987, pp. 247-251.
- [Conn94] J. H. Connolly and E. A. Edmonds.
«*CSCW and Artificial Intelligence*».
John H. Connolly and Ernest A. Edmonds (eds), Springer-Verlag, London, 1994.
- [Cout90] J. Coutaz.
«*Interfaces Homme-Ordinateur: Conception et Réalisation*».
Bordas-Dunod, Paris, 1990.
- [Crow90] T. Crowley et al.
«*MMConf: An Infrastructure for Building Shared Multi-media Applications*».
In Proceedings of the Third Conference on Computer-Supported Cooperative Work, Los-Angeles, California, October 8-10. ACM, New York, 1990.
- [Curt89] B. Curtis.
«*Organizational Computing, Coordination and Collaboration: Theories and Technologies for Computer-Supported Work*».
1989.

-
- [Dani86] T. Danielson, U. Panoke-Babatz, et al.
«*The AMIGO Project: Advanced Group Communication Model for Computer-based Communication Environment*»
In Proceedings of CSCW' 86. Austin, Texas-USA, December, 1986.
- [DeCi86] F. De Cindio, G. De Michelis et al.
«*CHAOS as a Coordinating Technology*»
In Proceedings of CSCW' 86. Austin, Texas-USA, December, 1986.
- [DeGo91] P. Degoulet, M. Fieschi.
«*Traitement de l'Information Médicale: Méthodes et Applications Hospitalières*»
Masson éditeur, 1991.
- [DeGo94] P. Degoulet, M. Fieschi.
«*Informatique Médicale*»
Masson éditeur, 1994.
- [Dery94] A. Derycke.
«*Le CSCW au delà de l'IHM: Taxonomie et Dimension Sociale*»
Présentation réalisée aux journées de groupe de travail GT SCOOP (GT8) du PRC Communication Homme-Machine, 13-14 juin, 1994, Lyon.
- [Dewa92] P. Dewan et al.
«*Critical Issues: User Interfaces to Distributed Systems*»
Engineering for Human Computer Interaction, J. Larson and C. Unger Eds., Elsevier Science Publishers B.V. (North-Holland), IFIP, Panel, 1992.
- [Digi93] DIGITAL Equipment Corporation.
«*Workflow at the Top of Desktop Integration*»
Technology Exchange Program-TEP, by Ralf Hausser, Digital Hannover, Germany, 1993.

- [Elli91] C. A. Ellis, S.J. Gibbs and G.L. Rein.
«*Groupware: Some Issues and Experiences*»
Communications of the ACM. Vol. 34, N°1, January 1991.
- [Enge63] Douglas Engelbart.
«*A Conceptual Framework for the Augmentation of Man's Intellect*»
In *Vistas in Information Handling*, Vol. 1 (P. Howerton, Ed), Spartan Books, Washington, DC, 1963, pp. 1-29.
- [Enge82] Douglas Engelbart.
«*Towards High-Performance Knowledge Workers*»
OAC'82, AFIPS Press.
- [Fish88] R. Fish, R. Kraut, M. Leland and M. Cohen.
«*Quilt: A Collaborative Tool for Cooperative Writing*»
In *Proceedings of the Conference on Office Information Systems*. Palo Alto, California, March 23-25. ACM, New York, 1988, pp 30-37.
- [Flor88] F. Flores, M. Graves, B. Hartfield and T. Winograd.
«*Computer systems and the design of organization interaction*»
ACM Trans. Office Information Systems, 6, 2, April 1988, 153-172.
- [Geer94] Wiebe Geertsma.
«*The Formalization of the Action Workflow Approach*»
Master's Thesis. Design Methodology Research Group, University of Twente, The Netherlands, 1994.
- [Gold84] A. Goldberg, D. Robson.
«*Smalltalk-80: The Language and its Implementation*»
Addison-Wesley Eds., 1984.

-
- [Gree91] S. Greenberg.
«*Computer-Supported Cooperative Work and Groupware: An Introduction to the Special Issues*»
International Journal of Man-Machine Studies. 1991, Vol. 34, pp. 133-141.
- [Grei86] I. Greif, R. Seliger and W. Weihl.
«*Atomic Data Abstractions in a Distributed Collaborative Editing System*»
In Proceedings of the 13th Annual Symposium on Principles of Programming Languages, St. Petersburg, Flo., January 13-15. ACM, New York, 1986, pp 160-172.
- [Grei88] I. Greif.
«*Computer-Supported Cooperative Work: A Book of Readings*»
Edited by I. Greif, Morgan Kaufmann Publishers, Inc. San Mateo, CA, 1988.
- [Grud88] J. Grudin.
«*Why CSCW Applications Fail: Problem in the Design and Evaluation of Organizational Interfaces*»
In Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'88). Portland, September 26-29, 1988, pp. 85-93.
- [Grud91] J. Grudin.
«*Obstacles to User Involvement in Software Product Development With Implications for CSCW*»
In Computer-Supported Cooperative Work and Groupware. Saul Greenberg ed., Academic Press, London 1991, pp. 313-330.
- [Grud94] J. Grudin.
«*Computer-Supported Cooperative Work: History and Focus*»
in IEEE Computer. Vol. 27: 5, May 1994, pp. 19-26.
- [Hahn91] U. Hahn, M. Jarke, K. Kreplin et al.
«*CoAuthor: A Hypermedia Group Authoring Environment*»
In J. M. Bowers and S. D. Benford (eds): Studies in Computer-Supported Cooperative Work. Theory, Practice and Design, North-Holland, 1991, pp. 79-100.

- [Hill93] David R. C. Hill.
«*Analyse Orientée Objets & Modélisation par Simulation*»
Addison-Wesley Eds. Paris, 1993.
- [Hodg92] H. Hodge.
«*Traitement des données et Stratégie de Gestion Hospitalière*»
J. Demongeot, P. Le Beux, et G. Weil (rédacteurs), *Informatique et Santé*, Vol. 11,
1992, pp. 27-32, Springer-Verlag, Paris.
- [Hogg85] J. Hogg.
«*Intelligent Message System*»
In Office Automation, D. Tschritzis, Ed. Springer Verlag, New York, 1985, pp. 113-
133.
- [Holt88] A. W. Holt.
«*A New Language for the Study and Implementation of Coordination*»
ACM Trans. Office Information Systems. 6, 2, April 1988, pp 109-125.
- [Inba79] M. Inbar.
«*Routine Decision-Making*»
The Future of Bureaucracy Beverly Hills: Saje, 1979.
- [Inga] D. Ingalls.
«*The Smalltalk-76 Programming System Design and Implementation*»
In Proceedings of the Fifth Annual ACM Symposium on Principles of Programming
Languages, ACM Press.
- [ISAR94a] R. Beuscart, B. Bossard, J. Brender, P. McNair, J. Talmon, P. Nikänen, M. Demeester.
«*AIM-ISAR: Methodology for Assessment of the Integration Process*»
In Public Report, Integration System ARchitecture, Deliverable 3, Work Package 1,
19 September 1994.

-
-
- [ISAR94b]** R. Beuscart, A. Dubos, N. Blassel, D. Roussel et al.
«*AIM-ISAR: Evaluation of the AIM Projects to be Integrated and Agreement on Detailed Objectives*»
In Public Report, Integration System ARchitecture, Deliverable 1, Work Package 2, 31 May 1994.
- [Joha84]** R. Johansen.
«*Teleconferencing and Beyond: Communications in the Office of the Future*»
McGraw Hill, New York, 1984.
- [Joha88]** R. Johansen.
«*Groupware, Computer Support for Business Teams*»
New York: The Free Press (McMillan Inc.), 1988.
- [John82]** P. Johnson-Lentz and T. Johnson-Lentz.
«*Groupware: The Process and Impacts of Design Choices*»
In Computer-Mediated Communication Systems: Status and Evaluation. E. B. Kerr, and S. R. Hiltz, Academic Press, New York, 1982.
- [Joos93]** Stef Joosten and Sjaak Brinkkemper.
«*Modelling of Working Groups in Computer Supported Cooperative Work*»
Technical Report, University of Twente, July 1993.
- [Joos94]** Stef Joosten, Gaston Aussems, Matthijs Duitshof et al.
«*Wa-12: An Empirical Study about the Practice of Workflow Management*»
Technical Report, University of Twente, July 1994. Research Monograph.
- [Kais87]** G.E. Kaiser, S. M. Kaplan and J. Micallef.
«*Multiuser, Distributed Language-Based Environments*»
IEEE Softw. 4, 6, November 1987, pp. 58-67.

- [Kars93] A. Karsenty, M. Beaudoin-Lafon.
«*An Algorithm for Distributed Groupware Applications*»
In Proceedings of the 13th IEEE International Conference on Distributed Computing Systems ICDCS'93. PA-Pittsburgh, USA, May 25-28, 1993.
- [Kies85] S. Kiesler, D. Zubrow, A. Moses and V. Geller.
«*Affect in Computer-Mediated Communication: An Experiment in Synchronous Terminal-To-Terminal Discussion*»
Human-Computer Interaction, Vol, 1, 1985, pp. 77-104.
- [Lela88] M. P. D. Leland, R. S. Fish and R. E. Kraut.
«*Collaborative Document Production Using QUILT*»
In Proceedings of the First Conference on CSCW'88. Portland, USA, September 26-28, ACM Press, New York, 1988, pp. 206-215.
- [Lewi88] B. T. Lewis and J. D. Hodges.
«*Shared Books: Collaborative Publication Management for an Office Information System*»
In Proceedings of the Conference on Office Information Systems. Palo Alto, California, March 23-25. ACM, New York, 1988, pp 197-204.
- [Lind90] P. Lindgreen (ed).
«*A Framework of Information Systems Concepts*»
Interim Report from the IFIP WG 8.1 task group FRISCO, 1990.
- [Loch88] F. H. Lochovsky, J. S. Hogg, S. P. Weiser and A. O. Mendelzon.
«*OTM: Specifying Office Tasks*»
In Proceedings of the Conference on Office Information Systems. Palo Alto, California, March 23-25. ACM, New York, 1988, pp 46-53.
- [Lotu93] LotusNotes®.
«*LotusNotes: Un Système permettant de Gérer la Mémoire de l'Entreprise*»
Rapport technique de LotusNotes®, CEDRIA, Paris, 1993.

-
-
- [Malo85] T. W. Malone.
«*Designing Organizational Interfaces*».
In the Proceedings of the CHI'85: Human Factors in Computing Systems, San Francisco, USA, April 14-18, 1985, pp. 66-71.
- [Malo86a] T. W. Malone, K. R. Grant, K.-Y. Lai, R. Rao and D. Rosenblitt.
«*Semi-Structured Messages are Surprisingly Useful for Computer-Supported Coordination*».
In the Proceedings of the Conference on Computer-Supported Cooperative Work, Austin-Texas, USA, December 1986.
- [Malo86b] T. W. Malone, K. R. Grant and F. A. Tubak.
«*The Information Lens: An intelligent system for information sharing in organisations*».
In the Proceedings of the CHI'86 Conference on Human Factors in Computing Systems, Boston-Massachusetts, Apr. 13-17 1986. ACM, New York, 1986, pp. 1-8.
- [Malo90] Thomas W. Malone and Kevin Crowston.
«*What is Coordination Theory and How It Can Help Design Cooperative Work Systems?*».
In the Proceedings of the Conference on Computer-Supported Cooperative Work, Los Angeles, USA, October 1990, pp. 357-370.
- [Mars92] R. T. Marshak.
«*Requirements for Workflow Products*».
In David D. Coleman, editor, *Groupware'92*, pp. 281-285, San Mateo, CA, USA, 1992. Morgan Kaufmann Publishers Inc.
- [McCa91] J. C. McCarthy, V. C. Miles and A. F. Monk.
«*An Experimental Study of Common Ground in Text-Based Communication*».
In Proceedings of the Human Factors in Computing Systems Conference, CHI'91, S. Roberston, G. Olson and J. Olson (eds), New Orleans, ACM, New York, 1991, pp. 209-215.

- [Medi92] Raül Medina-Mora.
«*ActionWorkflow Technology and Applications for Groupware*».
In David D. Coleman editors, *Groupware'92*, pp 165-167, San Mateo, CA, USA, 1992.
Morgan Kaufmann Publishers Inc.
- [Mint73] Henry Mintzberg.
«*The Nature of Managerial Work*».
Harper and Row, NY, 1973.
- [Mint79] Henry Mintzberg.
«*The Structuring of Organizations*».
Prentice-Hall, NY, 1979.
- [Mint83] Henry Mintzberg.
«*The Structure in Fives*».
Prentice-Hall, NY, 1983.
- [Mull86] S. J. Mullender and A. S. Tanenbaum.
«*The Design of a Capability-Based Distributed Operating System*».
The Computer Journal, Vol: 29, N°: 4, 1986, pp 289-299.
- [Newt85] Newton.
«*Technology and Cooperative Work Labour Among the Orokaiva*».
Manking, Vol: 15, 1985, pp. 214-222.
- [NSF89] NSF-IRIS.
«*A report by the NSF-IRIS review Panel for Research on Coordination Theory and Technology*».
NSF Forms and Publications Unit, 1989.
- [Oppe88] S. Opper.
«*A Groupware Toolbox*».
Byte, December 1988.

-
-
- [Otal90] OTALIA.
«*Démarche Utilisateurs pour la Conception du Système d'Information OTALIA*».
CHRU de Lille, février 1992.
- [Proc94] R. Procter, A. McKinlay R. Woodburn and O. Masting.
«*Coordination Issues in Tools for CSCW*».
pp. 119-137, 1994.
- [Rodd91] Tom Rodden and Gordon Blair.
«*CSCW and Distributed Systems: The problem of Control*».
In the Proceedings of the Second European Conference on Computer-Supported Cooperative Work, Bannon L. Robinson M. & Schmidt K. (Editors). September 25-27, 1991, Amsterdam, The Netherlands.
- [Rose94] D. Rosenberg and C. Hutchison.
«*Design Issues in CSCW*».
Duska Rosenberg and Chris Hutchison (eds), Springer-Verlag, London, 1994.
- [Sari85] S. Sarin and I. Greif.
«*Computer-Based Real-Time Conferencing Systems*».
IEEE Comput. Vol. 18, N°10, October 1985, pp 33-45.
- [Scha90] T. Schal and B. Zeller.
«*A Methodological Approach to Computer-Supported Cooperative Work*».
In Proceedings of the fifth European Conference on Cognitive Ergonomics. Urbino, September 3-6, 1990.
- [Sche93] A. L. Scherr.
«*A New Approach to Business Processes*».
IBM Systems Journal 32:1, pp. 80-98, 1993.
- [Sche90] J. R. Scherrer, R. Baud, D. Hochstrsser and O. Ratib.
«*An Integrated Hospital Information System*».
Medical Computing, 1990, Vol. 7, pp. 81-89.

- [Schm91] K. Schmidt.
«*Cooperative Work: A Conceptual Framework*».
In *Distributed Decision Making: Cognitive Models for Cooperative Work*.
Rasmussen Leplat eds, pp. 76-110, 1991.
- [Sear69] J. R. Searle.
«*Speech Acts Theory: An Essay in the Philosophy of Language*».
Cambridge University Press, New York, 1969.
- [Sear72] J. R. Searle.
«*Les Actes de Langage: Essai de Philosophie du Langage*»
Hermann, Paris, 1972.
- [Sear86] J. R. Searle.
«*Expression and Meaning: Study in the Theory of Speech Acts*».
Cambridge University Press, New York, 1986.
- [Shoc82] J. F. Shoch, Y. K. Dalal, D. D. Redell and R. C. Crane.
«*Evolution of the Ethernet Local Computer Network*»
IEEE Computer, August 1982. pp. 10-26.
- [Shor76] E. H. Shortliffe and L. E. Perreault.
«*Computer-Based Medical Consultations: MYCIN*»
New York: Elsevier/North-Holland, 1976.
- [Shor90] E. H. Shortliffe and L. E. Perreault.
«*Medical Informatics: Computer Applications in Health Care*»
Edouard H. Shortliffe and Leslie E. Perreault (eds), Addison-Wesley Publishing Inc.,
1990.
- [Simo57] H. A. Simon.
«*Model of Man*»
New York: John Wiley, 1957.

-
- [Spec89]** A. Spector.
«*Distributed Transaction Processing Facilities*»
In Mullender, S. (Ed): *Distributed Systems*, Addison-Wesley, New York, 1989.
- [Stef87]** M. Stefik, D. G. Bobrow, G. Foster, S. Lanning and D. Tatar.
«*WYSIWIS revised: Early Experiences with Multiuser Interfaces*»
ACM Trans. Office Information Systems, 5, 2, April 1987, pp. 147-186.
- [Tata91]** B. G. Tatar, G. Foster and D. G. Bobrow.
«*Design for Conversation: Lessons from COGNOTER*»
In *Computer-Supported Collaborative Work and Groupware*, edited by Saul Greenberg, Harcourt Brace Jovanovich, Publishers, 1991, pp. 55-80.
- [Wata90]** K. Watabe, S. Sakata K. Maeno et al.
«*Distributed Multiparty Desktop Conferencing System: MERMAID*»
In *Proceedings of CSCW'90*, Los Angeles, CA, October 7-10, 1990, ACM Press.
- [Wilb88]** S. B. Wilbur, R. E. Young.
«*The COSMOS Project: A Multi-Disciplinary Approach to Design of Computer Supported Group Working*».
in R. Speth (Ed): *EUTECO' 88: Research into Networks and Distributed Applications*, Vienna, Austria, April 20-22, 1988.
- [Wino87a]** Terry Winograd.
«*A Language/Action Perspective on the Design of Cooperative Work*».
Human-Computer Interaction, vol. 3, 1987.
- [Wino87b]** T. Winograd and F. Flores.
«*Understanding Computers and Cognition: A New Foundation for Design*».
Addison Wesley Publishing Company Inc., New York, 1987.

- [Whit94] Thomas E. White and Layna Fischer.
«*New Tools for New Times: The Workflow Paradigm. The impact of information on business process reengineering*».
Edited by T.E. White and Layna Fischer, Future Strategies Inc., 1994.
- [Yous94a] F. Yousfi, J-M. Geib and R. Beuscart.
«*A New Architecture for Supporting the Group Work in Health Care Domain*».
In proceedings of the 3rd IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET'ICE 94), WV-Morgantown, USA, April 1994.
- [Yous94b] F. Yousfi, R. Beuscart and J-M. Geib.
«*Workflow: un Concept Théorique pour la Coordination et la Modélisation des Entreprises*».
Workshop GANYMEDE sur le travail coopératif et la communication avancée.
Publication Interne au LIFL n° 159, Lille, juin 1994.
- [Yous95a] F. Yousfi, R. Beuscart and J-M. Geib.
«*PLACO: un Système Coopératif Multi-Agents dédié à la Gestion du Workflow dans les USI*».
Workshop GANYMEDE sur le travail coopératif et la communication avancée: Le Multi-Agents. Publication Interne au LIFL n° 163, Lille, février 1995.
- [Yous95b] F. Yousfi, R. Beuscart and J-M. Geib.
«*Workflow: A New Modelling Concept in Critical Care Units*».
In proceedings of the 8th World Congress on Medical Informatics (MED'INFO 95), Vancouver, CANADA, July 1995.
- [Yous95c] F. Yousfi, R. Beuscart and J-M. Geib.
«*PLACO: A New Cooperative Architecture Based on Workflow for Medical Field*».
In proceedings of the 2nd International Conference on Concurrent Engineering (CE'95): Research and Applications, Washington, DC Area, USA, August 1995.

[Yous95d] F. Yousfi, R. Beuscart and J-M. Geib.

«PLACO: A Cooperative Architecture for Managing Workflow in Critical Care Units».

In proceedings of the 19th Annual Symposium on Computer Applications in Medical Care (SCAMC'95), New Orleans, LA, USA, October-November 1995.

*PARTIE
DES
ANNEXES*

Annexe A

Graphe d'Héritage de la Solution PLACO

GRAPHE D'HERITAGE COMPLET

OBJECT

ObjetPlaco

```

| ObjetPersonne
|   | ActeurSoins
|   |   | Groupe
|   |   |   | GrpInfirmieres
|   |   |   | GrpMedecins
|   |   |   | GrpMixte
|   |   | Personne
|   |   |   | Infirmiere
|   |   |   |   | InfirmiereChef
|   |   |   | Medecin
|   | Malade
| ObjetPrescrit
|   | ActeMedical
|   | ExamenMedical
|   | Medicament
| PostIt
|   | PostitCoordination
|   |   | PostitAction
|   |   |   | PostitActionAdmin
|   |   |   | PostitActionEval
|   |   |   | PostitActionNull
|   |   |   | PostitActionPlanif

```


- | | | | *PostitActionPrescript*
- | | | | *PostitComment*
- | | | | *PostitRequest*
- | | | | *PostitDelegation*
- | | | | | *PostitAcceptDelegation*
- | | | | | *PostitRefuseDelegation*
- | | | | | *PostitRequestDelegation*
- | | | | *PostitClarification*
- | | | | | *PostitAcceptClarification*
- | | | | | *PostitRefuseClarification*
- | | | | | *PostitRequestClarification*
- | | **PostitFree**
- | | **PostitWorkflow**
- | | | | *PostitAdministrat*
- | | | | *PostitEchec*
- | | | | *PostitEvaluat*
- | | | | *PostitPlanif*
- | | | | *PostitPrescript*
- | **WorkflowItem**
- | | **Administration**
- | | | | *AdminActeMedical*
- | | | | *AdminExamenMedical*
- | | | | *AdminMedicament*
- | | **Evaluation**
- | | | | *EvalActeMedical*
- | | | | *EvalExamenMedical*
- | | | | *EvalMedicament*
- | | **Planning**
- | | | | *PlanActeMedical*
- | | | | *PlanExamenMedical*
- | | | | *PlanMedicament*
- | | **Prescription**
- | | | | *PrescriptActeMedical*
- | | | | *PrescriptExamenMedical*
- | | | | *PrescriptMedicament*

Exemples d'Interfaces du Prototype

Pour mieux comprendre l'enchaînement des phases du cycle workflow et les messages générés, nous reprenons le scénario récurrent que nous avons détaillé dans les chapitres précédents, à savoir la prescription de **Dobutrex** pour un patient.

Nous décrirons, via des copies d'écran, le déroulement des actions effectuées sur l'objet prescrit **Dobutrex** dans le cycle de workflow.

Quatre acteurs sont impliqués dans le cycle. Il s'agit des médecins Cécile et Marc, et des infirmiers Yves et Valérie.

Le médecin Cécile effectue une prescription pour le patient Eric. Ensuite, l'infirmière Valérie réalise la planification de ce médicament. Quant à l'infirmier Yves, il s'occupera de l'administration effective du médicament au patient. Enfin, le médecin Marc réalise l'évaluation de la prescription et émet une décision selon les résultats obtenus.

Fonctionnalités offertes par le gestionnaire des «post-it»

Liste des patients admis en unité.

The screenshot shows a window titled "User Window <Médecin5> <Cécil>". At the top, it displays "Patient Posta" and "Patient's record List". The record list includes: Malade1_Eric, Malade7_Sylvain, Malade2_Philippe, Malade3_Jean, Malade6_Marie, Malade4_Josianne, and Malade5_Nadia. To the right, user information is shown: "Profil: <Médecin> Nom: <Médecin5> Prénom: <Cécile> Code: <10>". Below the record list is a "Post-it List" which is currently empty, displaying "No Post-it for the Selected Patient". A vertical toolbar on the right contains buttons for: Go to... (go to the selected patient's record), New... (create a new post-it), Reply (reply to the selected post-it sender), Edit (edit the selected post-it), Save (Save the selected post-it), Load... (Load the saved Post-its), Delete (delete the selected post-it), and Refresh (refresh the post-it list). At the bottom left are "Delete All" and "Display All" buttons. At the bottom right is a "Quit" button. A "Message Bar (Read Only)" at the bottom left contains the text: "Sorry! Any new Post-it for this Patient!".

«Post-its» de Coordination
Liste des «post-it» de coordination reçus par le médecin Cécile. Ces «post-it» annoncent les actions à exécuter sur les objets prescrits du patient Eric.

Fenêtre d'utilisateur
Elle contient la liste des patients admis en unité, et les post-it envoyés à l'utilisateur connecté (le médecin Cécile).

Liste des «post-it» de workflow relatifs à l'objet prescrit sélectionné.

Détail du «post-it» de workflow sélectionné. Le «post-it» sélectionné spécifie la prochaine action à exécuter (i.e. planification).

Liste des objets prescrits pour le patient Eric.

2. PLACU - Workflow Items. User: <Médecin5_Cécile>

File ObjetPrescrit Prescription Planning Delivery Evaluation

Patient Informations

Nom: <Meledel> Prénom: <Eric> Code: <1> Age: <35>

Objets Prescrits

Dobutrex made by <Médecin5>

Liste des Post-It

Post-It de Prescription: envoyé <le 04/01/96 à 11:33:50>

Type: <Post-It de Prescription>
Date de Création: <le 04/01/96 à 11:33:50>
Emetteur: Médecin5_Cécile
Sujet: <Phase de Prescription>
Texte: < le PLANNING est à faire>

Détail du Post-It Sélectionné

EVOLUTION DE LA PRESCRIPTION DANS LE CYCLE WORKFLOW

Prescription	Administration	Evaluation
Type de Prescription: Objet Médicament Nom: Dobutrex Nom DCI: SAP Date de Prescription: le 04/01/96 à 11:33:50 en ... de Fréquence: 2 Dose: 250 en ml Durée de Traitement: 1 J	No Delivery	No Evaluation

«Post-its» de Workflow

Le gestionnaire de «post-it» a créé un «post-it» de workflow pour rendre compte de l'état de l'objet prescrit Dobutrex dans le workflow.

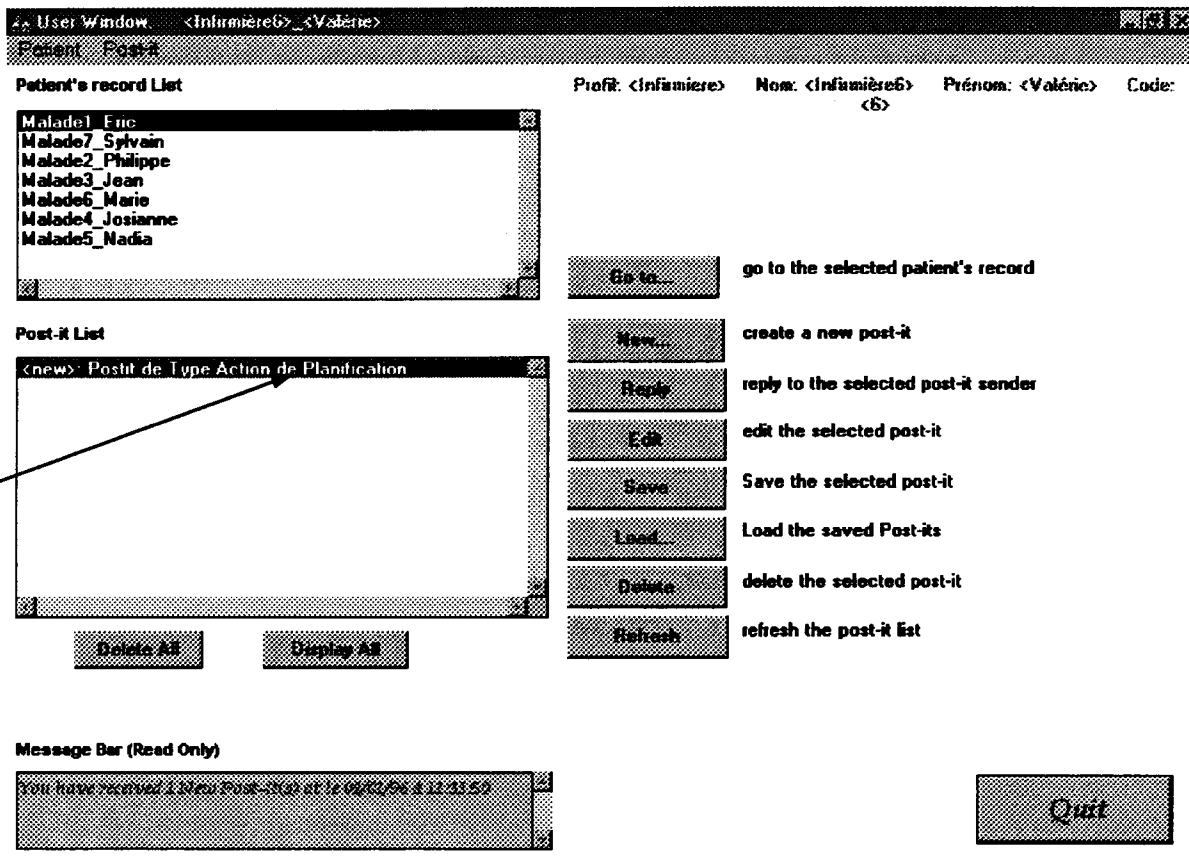
L'administration de Dobutrex au patient est en attente. Elle doit être exécutée après la planification

L'évaluation de Dobutrex s'effectue après l'administration pour rendre compte de l'état du patient Eric.

Gestionnaire de Workflow

Le médecin Cécile a prescrit le médicament Dobutrex pour le patient Eric en spécifiant les informations nécessaires à la planification et l'administration.

La planification de Dobutrex est en attente.



«Post-it» de Coordination
L'infirmière Valérie a reçu un «post-it» de type action de planification. Ce dernier lui annonce l'action de planification à effectuer pour le patient Eric.

Fenêtre d'utilisateur
L'infirmière Valérie dispose des mêmes fonctionnalités (concernant les «post-it») qu'un médecin.

Gestionnaire de Workflow (Patient Eric)
 Fenêtre indiquant l'évolution des objets
 prescrits dans le workflow pour
 l'infirmière Valérie.

PLACO - Workflow Items. User: <Infirmiere6_Valerie>

File ObjetPrescrit Prescription Planning Delivery Evaluation

Patient Informations

Nom: <Malade1> Prénom: <Eric> Code: <1> Age: <35>

Objet Prescrit: Dobutrex made by <Medecin5>

Liste des Post-its

Détail du Post-it Sélectionné

Post-it créé dans la Phase de Planification envoyé le 04/01/96 à 11:36:13

Type: <Post-it créé dans la Phase de Planification>
 Date de Création: <le 04/01/96 à 11:36:13>
 Emetteur: Infirmiere6_Valerie
 Sujet: <Phase de Planification>
 Texte: <l'ADMINISTRATION est à faire>

EVOLUTION DE LA PRESCRIPTION DANS LE CYCLE WORKFLOW

Prescription	Planification
Type de Prescription: Objet Médicament Nom: Dobutrex Nom DCI: SAP Date de Prescription: le 04/01/96 à 11:33:50 en ... de Fréquence: 2 Dose: 250 en ml Durée de Traitement: 1 J	Type: Objet Médicament Date de Planning: le 04/01/96 à 11:36:13 Acteur: Infirmiere6_Valerie Horaires de Prise: 24 H: 12 H:
Administration	Evaluation
No Delivery	No Evaluation

«Post-it» de Workflow

Un nouveau «post-it» a été créé annonçant l'action suivante à exécuter sur Dobutrex (i.e. administration). L'ancien «post-it» est détruit automatiquement par le gestionnaire de «post-it» puisqu'il est obsolète. L'action de planification, qui lui est associée, est déjà effectuée.

L'infirmière Valérie prend connaissance des informations obtenues lors de la phase précédente, et effectue la planification de Dobutrex pour le patient Eric.

L'action d'administration de Dobutrex est en attente. Le prochain acteur dispose de toutes les informations relatives à la prescription et à la planification de ce médicament.

L'action d'évaluation doit être exécutée après l'administration de Dobutrex au patient Eric.

Gestionnaire de Workflow

Fenêtre permettant à l'infirmier Yves de prendre connaissance de l'évolution de l'objet Dobutrex dans le cycle workflow.

The screenshot shows a software window titled 'PLACU - Workflow Items. User: <Infirmier2_Yves>'. The window has a menu bar with 'File', 'Objet/Prescrit', 'Prescription', 'Planning', 'Delivery', and 'Evaluation'. Below the menu bar, there are several sections:

- Patient Informations:** Nom: <Maledel>, Prénom: <Eric>, Code: <1>, Age: <35>.
- Objets Prescrits:** A list containing 'Dobutrex made by <Médecin5>'.
- Liste des Post-It:** A list of 'Post-it d'Administration' entries, all dated '04/01/96 à 11:37:30'. One entry is highlighted.
- Détail du Post-It Sélectionné:** Type: <Post-it d'Administration>, Date de Création: <le 04/01/96 à 11:37:30>, Emetteur: Infirmier2_Yves, Sujet: <Phase d'Administration>, Texte: < EVALUATION est à Faire >.
- EVOLUTION DE LA PRESCRIPTION DANS LE CYCLE WORKFLOW:** A central section divided into four quadrants:
 - Prescription:** Type de Prescription: Objet Médicament, Nom: Dobutrex, Nom DCI: SAP, Date de Prescription: le 04/01/96 à 11:33:50 en ... de Fréquence: 2, Dose: 250 en ml, Durée de Traitement: 1 J.
 - Planification:** Type: Objet Médicament, Date de Planning: le 04/01/96 à 11:36:13, Acteur: Infirmière6_Valérie, Horaires de Prise: 24 H; 12 H.
 - Administration:** Type: Objet Médicament, Date de Administration le 04/01/96 à 11:37:30, Acteur: Infirmier2_Yves, Prises Effectuées: 24 H. Administré: Ok par: Infirmier2_Yves avec décalage de: -20 ms; 12 H. Administré: Ok par: Infirmier2_Yves avec décalage de: 15 min.
 - Evaluation:** No Evaluation.

L'infirmier Yves a fini d'administrer le médicament Dobutrex au patient Eric. A chaque administration, il a la possibilité de générer un commentaire pour décrire la situation et/ou l'état du patient.

Un «post-it» d'administration est généré pour annoncer la prochaine étape à effectuer sur l'objet Dobutrex. L'ancien «post-it» de planification est détruit par le gestionnaire des «post-it».

L'action d'évaluation est en attente. Le prochain acteur doit donc prendre connaissance des informations spécifiées dans les trois dernières phases pour mener à bien sa tâche.

Fenêtre d'utilisateur

Le médecin Marc reçoit un «post-it» de coordination qui lui indique l'action à effectuer, en l'occurrence l'évaluation.

The screenshot displays a software window titled "User Window" for a doctor named Marc. It features a "Patient's record List" with names like Malade1_Eric and Malade7_Sylvain. A "Post-it List" shows a new note of type "Action d'Evaluation". A vertical toolbar on the right contains buttons for "Go to...", "New...", "Reply", "Edit", "Save", "Load", "Delete", and "Refresh", each with a corresponding description. At the bottom, a "Message Bar (Read Only)" shows a notification about a received post-it, and a "Quit" button is located in the bottom right corner.

User Window: <Médecin>_<Marc>

Patient Post-it

Patient's record List

Prof: <Médecin> Nom: <Médecin> Prénom: <Marc> Code: <1>

Malade1_Eric
Malade7_Sylvain
Malade2_Philippe
Malade3_Jean
Malade6_Marie
Malade4_Josianne
Malade5_Nadia

Go to... go to the selected patient's record

Post-it List

<new> Postit de Type Action d'Evaluation

New... create a new post-it

Reply reply to the selected post-it sender

Edit edit the selected post-it

Save Save the selected post-it

Load Load the saved Post-its

Delete delete the selected post-it

Refresh refresh the post-it list

Delete All Display All

Message Bar (Read Only)

You have received 1 new Post-it(s) at le 06/02/06 à 11:57:50

Quit

PLACD - Workflow Items, User: <Médecin1_Marc>

File: ObjetPrescrit Prescription Planning Delivery Evaluation

Patient Informations: Objets Prescrits

Nom: <Malade1> Prénom: <Eric> Code: <1> Age: <35> Dobutrex made by <Médecin5>

Liste des Post-its Détail du Post-it Sélectionné

Postit d'Administration envoyé le 04/01/96 à 11:37:30	Type: <Postit d'Evaluation>
Postit d'Administration envoyé le 04/01/96 à 11:37:30	Date de Création: le 04/01/96 à 11:49:20
Postit d'Evaluation envoyé le 04/01/96 à 11:49:20	Emetteur: Médecin1_Marc
	Sujet: <Phase d'Evaluation>
	Texte: < Cycle Workflow terminé>

EVOLUTION DE LA PRESCRIPTION DANS LE CYCLE WORKFLOW

Prescription	Planification
Type de Prescription: Objet Médicament Nom: Dobutrex Nom DCI: SAP Date de Prescription: le 04/01/96 à 11:33:50 en ... de Fréquence: 2 Dose: 250 en ml Durée de Traitement: 1 J	Type: Objet Médicament Date de Planning: le 04/01/96 à 11:36:13 Acteur: Infirmière6_Valérie Horaires de Prise: 24 H: 12 H:

Administration	Evaluation
Type: Objet Médicament Date de Administration le 04/01/96 à 11:37:30 Acteur: Infirmier2_Yves Prises Effectuées: 24 H, Administré: Ok par: Infirmier2_Yves avec décalage de: -20 m 12 H, Administré: Ok par: Infirmier2_Yves avec décalage de: 15 min	Type: Objet Médicament Date d'Evaluation: le 04/01/96 à 11:49:20 Acteur: Médecin1_Marc Résultat: Supprimer la Prescription

«post-it» de Workflow

Un «post-it» est créé pour annoncer la fin du cycle workflow. L'ancien «post-it» est détruit automatiquement puisqu'il est obsolète.

Fenêtre de Workflow

Le médecin Marc a effectué l'évaluation de l'objet prescrit Dobutrex. Il prend connaissance des informations créées lors des quatre phases. Le résultat de l'évaluation est la suppression du médicament.

Annexe C

*Spécimen de Feuilles de
Prescription et de
Surveillance en
Réanimation Polyvalente*



FEUILLE DE SURVEILLANCE

Service: BPVAIntervention: ↓Date: 27/06

HEURE		14	16	18	20	22	24	(1 ^h)/2	4	6	8	10	12	
HÉMODYNAMIQUE	T.A.													
	25													
	20													
	15													
	10													
	Temp.													
	Pouls													
	40	140												
	39	120												
	40	100												
37	080													
36	060													
PVC							18	14	14		12			
DIURÈSE							100	200	35		600			
PERTES	VC 50%						33				VC 50%			
	spno II						11/14	11/14	12/1		16			
	freq						16/16	16/16			149/min			
	S/G						50	50	50		50			
SOLUIES	SG 5% 500ml													
BILANS SANGUINS	TRZ 4x2													
	Treillem 200													
	Reprel 1 amp													
	Prillem 1 amp													
											DC 4" G, 20 l/min IC 500 l/min/m ² Sous less 12mg/h			



FEUILLE DE SURVEILLANCE

ETIQU

Service: RPA

Intervention:

Date: 24/6

HEURE		14	16	18	20	22	24	2	4	6	8	10	
HÉMODYNAMIQUE	T.A.	160/90	160/90	160/90	160/90	160/90	160/90	160/90	160/90	160/90	160/90	160/90	
	FC	100	100	100	100	100	100	100	100	100	100	100	
	CAP	10	10	10	10	10	10	10	10	10	10	10	
	PA	10	10	10	10	10	10	10	10	10	10	10	
	Temp.	37	37	37	37	37	37	37	37	37	37	37	
	Pouls	100	100	100	100	100	100	100	100	100	100	100	
	PVC	17	13	13	13	13	13	13	13	13	13	13	
	DIURÈSE												
	Ep. chg	E	E	E	E	E	E	E	E	E	E	E	E
	Mode	VC	VC	VC	VC	VC	VC	VC	VC	VC	VC	VC	VC
FiO2	40%	40%	40%	40%	40%	40%	40%	40%	40%	40%	40%	40%	
SpO2	94%	94%	94%	94%	94%	94%	94%	94%	94%	94%	94%	94%	
Resuena	16	16	16	16	16	16	16	16	16	16	16	16	
SOLUTÉS	SG	30%	50%										
	ZL	20%	25%										
	Multicad	2x 100ml											
		12 Elixir											
THÉRAPEUTIQUE	DOPN	250/14											
	Tops	4g x 2											
	Aspirin	1g x 2											
	Trifluor	400											
	Sulfate	2amp 12h											
	Ulipristal	2amp 16h											
	Acetamin	1amp 12h											
	Tapestral	2amp 16h											
BILANS SANGUINS	16 ^h GR 3,50												
	GB 14,3												
Na 131													
K 4,10													
Se 1,59													
Créat 51													
Fe 0,14													
Ph 7,31													
42													
96													
4.3													

L.F.B

ALBUMINE 4% 250 ML

LOT: 40140231
DLU: SEPT 97

L.F.B

ALBUMINE 4% 250 ML

LOT: 40140231
DLU: SEPT 97

L.F.B

ALBUMINE 4% 250 ML

LOT: 40125470
DLU: OCT 95

GC = 5,43

IC = 2,90

80ms 1000
14



FEUILLE DE SURVEILLANCE

ETIQUETTE

Service: Réa Pdy A

Intervention:

Date: 28/06/95

HEURE	14	16	18	20	22	24	2	4	6	8	10	12
T.A.					58							
PAP	54/20	54/13	74/24		60/24		62/21		65/21	12		12
CAP	32		2		(34)		31		5	5		5
	13	20	19		(12)		(12)		(14)	(16)		(15)
Temp.					36		36		36	36		36
Pouls					66		65		67	64		64
40					140		140		140	140		140
39					120		120		120	120		120
40					100		100		100	100		100
37					080		080		080	080		080
36					060		060		060	060		060
PVC	12	15	21		(18)		(17)		(20)	(18)		(13)
DIURÈSE	12cc	11cc			20cc		20cc		20cc	30		
SG	300	400			500		600		700	5		
lame	50	50			100		100		100	5		
Ileostomie					100		100		100	50		
					mode 12				VC	VC		
					F102 40%				40%	40%		
					Su10 14				14	14		
					Respirance 16				16	16		
SG	30%	500										
2L	20%	250										
Multic	2	100										
Nell	0	500										
Dobal	250	1/4										
Tazo	4	1/2										
Amikine	1/2	1/2										
Typhine	100											
Burine	500											
Sulfate	2	amp										
Cupromed	1	amp										
Chlorid	1											
Taxim	1	amp										
CG												

BILANS SANGUINS	8.158623	1.158658	9.158659
CP	8234855	0296072	0234825
	2242042	7234835	6451780
		5234870	



FEUILLE DE SURVEILLANCE

CENTRE HOSPITALIER DE HAUTEFERRIERES

Service: Réa Poly

Intervention:

Date: 1/07/95

HEURE	14	16	18	20	22	24	2	4	6	8	10	12
	T.A.											
25		7	7	3								
20			22									
15												
10			14/55									
Temp.	40	39	40	37	36							
Pouls	140	120	100	080	060							
PVC			15									
DIURÈSE			50									
CSG	300		5=0									
URINE												
SOLUTÉS	SG 300	500										
	P.L. 200	500										
	Vitamine	500										
	Na 400											
	DEBU 250	166										
	Togo 100	17										
	Amik 100	100										
	Ti 100	100										
	Air 100	100										
	Sulfate 100	100										
	Supp 100	100										
	SAD 500											
	Bizine 500											
BILANS SANGUINS												

L.F.B
ALBUMINE 4% 500 ML

LOT: 48140310
DLU: NOV 97



FEUILLE DE SURVEILLANCE

Service: Réa Poly

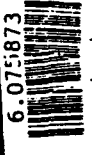
Intervention:

3/07

Date:

95

HEURE	14	16	18	20	22	24	2	4	6	8	10	12
	T.A.	100	114/5	107/6								
25	65	85										
20	68/29	68/29	54/23									
15	63	63	55									
10	19	12										
Temp. Pous	32											
40	140											
39	120											
40	100											
37	080											
36	060											
PVC	18	16	16									
DIURÈSE	0	11	15									
VC	150	150	150									
FIO ₂	35%	35%	35%									
SG												
SOLUTÉS	6.30% 500 TL 20% 250 Ventolin 250 6g KCl 24L											
THÉRAPEUTIQUE	Dabig 250mg 1h Tagamet 1amp Pains 40 2/24h Aillicor 100mg 1/24h Sulfato 2amp 1/24h Hypovitel 1amp											
BILANS SANGUINS	7.35 cord 4.7 364 1.9 137 5.25 14.7 4.25 4.4 0.8 Na 116 K: 3.8 debut dialyx (224) CR 3733 Hte 10 Hte 19.3 Play 80000 TP 757 TCA 98 T 30 CS ⊖ midialyx (OH) Na 132 K 340 cl 97 U 133 creat 35 CR 3713 Hte 9.3 Hte 29.2 Play 80000 Hte 133 K 33 cl 97 glu 1.41 creat 1.23 creat 32 CR 211.37 Hte 9 Hte 3 Play 80000 Hte 133											





FEUILLE DE SURVEILLANCE

Service: Réa Poly

Intervention:

4/07 Date:

HEURE	14	15	16	18	20	22	24	2	4	6	8	10	12
	HEURE	14	15	16	18	20	22	24	2	4	6	8	10
HÉMODYNAMIQUE	T.A.	98		97						105	97		
	25	96		97									
	Débit	4.0			4.5								
	20												
	15	4.6											
	10	4.6											
	Temp. Pous	38			36		36						
	40												
	39												
	40												
37													
36													
PVC	13			16		(17)		(18)		(13)	(15)		
DIURÈSE	0			50									
PERTES	SE			60									100
	VC						VC	VC	VC	VC	VC		
	FIO ₂	30%				30%		35%		35%	35%		
	surface / ml			1000									
SOLUTÉS	SC 30% 500												
	VL 20% 250												
	Vialine 250												
						3.80					250 mg	0/3h	
THÉRAPEUTIQUE	SC												
	Flud 500ml												
	Solu 250mg 16h												
	Paracet 1g 19h												
	Tax 4g 22/24h												
	Hypoval 10mg 18h												
BILANS SANGUINS	Suberona 2amp, 12h												
	<p>(14) après scutels</p> <p>GR 3,67 Hbe 117 Hc 112 Pneu G200 Lec 2: 800</p> <p>Genon Kip Plaque II</p>												

- & min - résiduel Amblin





FEUILLE DE SURVEILLANCE

Service: Réa. Poly

Intervention:

Date: 5/7

HEURE	HÉMODYNAMIQUE																	
	Temp		CAP		PVC		DIURESE		PERTES		SOLUTÉS		THÉRAPEUTIQUE					
	40	140	39	120	37	100	36	060	40	140	39	120	37	100	36	060		
14	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
16	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
18	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
20	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
22	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
24	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
2	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
4	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
6	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
8	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
10	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060
12	38	140	38	120	37	100	060	10	0	Adre	500ml	40	140	38	120	37	100	060

784066



784065



784067



784064



L.F.B
ALBUMINE 20% 100 ML
LOT: 20145262
DLU: JUIN 96

L.F.B
ALBUMINE 20% 100 ML
LOT: 20145262
DLU: JUIN 96

L.F.B
ALBUMINE 4% 500 ML
LOT: 40140211
DLU: SEPT 97

LAB 60
Eldo 80



FEUILLE DE SURVEILLANCE

Service: Rén. Poly.

Intervention:

Date: 06/07
45

HEURE		14	16	18	20	22	24	2	4	6	8	10	12
HÉMODYNAMIQUE	T.A.		57/58							35/6			
	25												
	20			128 54			160						
	15												
	10			53/54								116	
	Temp.												
	Pouls												
	40												
	39												
	40			86							80		
37						62/11							
36													
PVC							15			16	16		
DIURÈSE		E		E								E	
PERTES	VC FTO ₂ = 40%										6.89		
	V _D = 140 l/min										3.34		
	VK ₁	E		E			100			E			
	VK ₂	E		100			200			200			
	VK ₃	250		250			200			200			
SOLUTÉS	SG 30% 500ml												
	TL 20% 250ml												
	Vit. 250ml												
THÉRAPEUTIQUE	Dobut. 2.50 mg/6h												
	Adon 1.5 mg/6h												
	Tazac 40 x 2 / 12h												
	Vit. C 100mg												
	Augmentin 15 mg/6h												
	Bidac 20 mg / 12h												
BILANS SANGUINS	Ca 500												
	Coag. vit. C												

