UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

BRUNO BOUCHARD

LA MESURE DE LA SIMILARITÉ ENTRE LES POINTS DE VUE DE L'USAGER ET DE SON AGENT ARTIFICIEL À L'AIDE DE LA LOGIQUE TERMINOLOGIQUE

12 AOÛT 2003

Bibliothèque
Paul-Émile-Boulet

UQAC
Université du Québec
à Chicoutimi

Mise en garde/Advice

Afin de rendre accessible au plus grand nombre le résultat des travaux de recherche menés par ses étudiants gradués et dans l'esprit des règles qui régissent le dépôt et la diffusion des mémoires et thèses produits dans cette Institution, **l'Université du Québec à Chicoutimi (UQAC)** est fière de rendre accessible une version complète et gratuite de cette œuvre.

Motivated by a desire to make the results of its graduate students' research accessible to all, and in accordance with the rules governing the acceptation and diffusion of dissertations and theses in this Institution, the **Université du Québec à Chicoutimi (UQAC)** is proud to make a complete version of this work available at no cost to the reader.

L'auteur conserve néanmoins la propriété du droit d'auteur qui protège ce mémoire ou cette thèse. Ni le mémoire ou la thèse ni des extraits substantiels de ceux-ci ne peuvent être imprimés ou autrement reproduits sans son autorisation.

The author retains ownership of the copyright of this dissertation or thesis. Neither the dissertation or thesis, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

RÉSUMÉ

Les systèmes multi-agents actuels ne prennent pas en considération les problèmes de cohabitation entre un usager et son homologue agent virtuel agissant en son nom -on behalf-, tels que l'intégration de l'utilisateur dans la même boucle de réalisation d'une tâche commune - usager dans la boucle - et les conflits terminologiques liés à l'utilisation de termes différents pour décrire leurs opinions (points de vue) à propos d'une situation de résolution d'un problème. La cohabitation usager-agent nécessite donc une compréhention mutuelle, ce qui signifie que les deux entités devront être aptes à comparer leurs points de vue respectifs avant la prise d'initiatives. Par conséquent, la question qui se pose est la suivante: dans quelle mesure les deux points de vue (utilisateur et agent) peuvent ou non se rapprocher?

Ce travail de recherche vise à contribuer au processus général d'intégration d'un usager dans la boucle de réalisation d'une tâche conjointe. Il propose une approche de comparaison des points de vue dans un contexte de planification par initiatives mixtes. L'approche théorique proposée s'appuie sur la logique terminologique pour décrire les ontologies des points de vue de l'utilisateur et de l'agent. La méthode de comparaison des points de vue proposée dans ce mémoire permettra d'extraire une mesure de leur similarité, servant à prendre une décision sécuritaire.

Ce mémoire se veut une première phase d'un projet de recherche beaucoup plus large, visant le développement d'une approche générique d'intégration des points de vue. Il doit donc être considéré comme un pas en avant vers la réalisation de ce projet d'envergure ainsi qu'une contribution au domaine de la coopération entre un usager et son agent artificiel.

REMERCIEMENTS

En premier lieu, j'aimerais remercier mon directeur de recherche, M. Abdenour Bouzouane, pour son support, sa grande disponibilité, son suivi rigoureux de chacune des étapes de réalisation du projet et également pour son incroyable patience à mon égard. Sans lui, ce travail de recherche n'aurait sûrement jamais pu aboutir. Il faut également mentionner que son optimisme et sa bonne humeur ont grandement aidé à passer à travers les périodes les plus difficiles.

J'aimerais également remercier M. Sylvain Boivin, directeur de l'Unité d'Enseignement en Informatique et Mathématique, qui m'a poussé à aller de l'avant dans mes études de cycles supérieurs. Je dois également remercier M. Jean Rouette, directeur du Département d'Informatique et de Mathématique, ainsi que M. Richard Bouchard. Ils m'ont supporté dans toutes mes démarches et m'ont permis de m'initier au domaine de l'enseignement.

Dans un second temps, j'aimerais souligner le support irremplaçable de toute ma famille, mes parents ainsi que mes frères et ma sœur, qui ont cru en moi durant tout ce temps. Je dois un remerciement spécial à M. Normand Bouchard, qui m'a grandement aidé afin d'améliorer la qualité du français dans cet ouvrage.

Il ne faut pas oublier de souligner l'appui quotidien de tous mes collègues du groupe de recherche en informatique, sans qui la vie aurait été beaucoup plus monotone.

Finalement, je remercie ma compagne Annie-Claude Privé, qui a bien voulu m'aider et me supporter durant cette étape importante de ma vie et de ma future carrière.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

CHAPITRE 1

INTRODUCTION

L'utilisation massive des technologies de l'information dans divers domaines d'applications a modifié profondément la façon dont le travail est conçu, délégué et réalisé. On déléguera davantage de responsabilités à une entité informatique assurant le même rôle qu'un utilisateur, et agissant en son nom. Une des technologies les plus prometteuses pour réaliser la délégation de tâches est basée sur les systèmes multi-agents. Ces systèmes font partie de ce qu'il est convenu d'appeler l'Intelligence Artificielle Distribuée (IAD), branche relativement récente de l'intelligence artificielle [Ferber, 1995]. L'IAD s'intéresse à la conception d'agents artificiels capables de s'organiser pour accomplir collectivement les fonctionnalités qui leur sont demandées [Chaib-Draa et al., 1992] [Wooldridge, 2000]. Un agent est un programme capable de prendre ses propres décisions (autonome), de se déplacer sur le réseau, d'échanger de l'information avec d'autres agents, d'apprendre, etc. Par exemple, dans le cas du commerce électronique, on peut concevoir un agent réalisant les transactions et prenant le rôle d'une personne sur son lieu de travail. Dans le contexte « cohabitationiste » qui nous intéresse, l'usager et son agent peuvent coopérer pour réaliser conjointement une transaction commerciale, s'envoyer les uns aux autres des informations boursières, intervenir à tout moment pour prendre le contrôle de la tâche, prendre des initiatives, etc. Ainsi, la réalisation d'une tâche ne sera pas l'œuvre uniquement de l'utilisateur ou de l'agent, mais plutôt de l'ensemble - usager dans la boucle -, et dans lequel ils auront les mêmes possibilités de proposer, de suspendre, de refuser et d'arrêter l'autre [Dautenhahn, 2001].

En fait, cette cohabitation hybride est à la fois riche et complexe, du fait que l'usager et l'agent sont amenés non seulement à s'accorder sur les différents niveaux de réalisation de la tâche à l'intérieur d'une même boucle de résolution, mais aussi à gérer l'initiative mixte « qui contrôle qui? ». Une initiative signifie prendre une décision sur ce qu'on doit faire. Ceci n'implique pas que l'action préconisée sera exécutée par celui qui a pris l'initiative [Cesta et al., 1999]. Une initiative est mixte si elle peut être engendrée à n'importe quel moment par l'utilisateur ou par l'agent. Afin d'assurer une coopération usager-agent efficace et sécuritaire, particulièrement dans une situation de crise ou à risque, il est nécessaire pour eux d'être aptes à se comprendre et donc d'être aptes à comparer leurs points de vue respectifs d'une même situation [Falcone et Castelfranchi, 2001].

L'une des problématiques majeures de cette cohabitation concerne la divergence des points de vue causée par les différences terminologiques qui ne sont pas nécessairement conflictuelles [Tessier et al., 2001][Bouzouane et Bouchard, 2003]. Cette divergence peut être vue comme deux interprétations différentes pour une même réalité. La différence est due à l'hétérogénéité sémantique du même concept. En d'autres termes, le même concept qui est décrit par une terminologie différente (nom, domaine, contraintes, structure, contenu). Par exemple, lors d'une transaction commerciale, l'usager peut utiliser le terme *«contacter fournisseur »* dans son plan d'action pour contacter un certain fournisseur de produits et l'agent agissant en son nom élabore un plan d'actions similaire mais en utilisant un ou plusieurs termes différents tels que « rechercher grossiste », « négocier paiement », etc., ce qui rend complexe la comparaison de leurs points de vue. Cette disparité peut être

due aux transferts de connaissances et aux habilités d'apprentissage. Initialement, on suppose que ces deux entités ont le même savoir-faire, mais elles peuvent évoluer différemment dans le temps.

L'objectif général de notre travail de recherche est de répondre à la question suivante : peut-on élaborer un modèle « computationnel » de comparaison des points de vue divergents avant la prise de contrôle d'une tâche? D'une façon plus précise, notre premier objectif consiste à redéfinir en termes formels la comparaison des points de vue en s'appuyant sur la relation de subsomption de la logique terminologique [Nebel 1995] [Baader et al 2003]. Cette logique est une extension des réseaux sémantiques permettant la représentation des connaissances [Minsky, 1974] [Brachman et Schmolze, 1985] [McGregor, 1991] [Schmidt, 2000]. La subsomption est une méthode d'inférence pour la classification de concepts dans une ontologie. Une ontologie est une structure formelle organisée sous la forme d'une hiérarchie permettant de représenter d'une façon explicite la sémantique de l'information structurée ou semi-structurée relative à un domaine [Davies et al., 2003]. Comme deuxième objectif, nous visons principalement à étendre la relation de subsomption vers une relation pondérée par des heuristiques permettant de mesurer la crédibilité d'une initiative mixte basée sur la similarité des points de vue, dans le but de garantir une prise sécuritaire du contrôle de la tâche commune. Enfin, notre dernier objectif consiste à valider l'approche proposée sur un cas concret qui concerne le commerce électronique.

Pour atteindre ces objectifs, nous avons mené plusieurs investigations. La première porte sur une revue littéraire détaillée à propos des méthodes de coopération existantes dans le domaine des systèmes mutli-agents, telles que les plans partagés [Crosz et Kraus, 1998], qui se veut une méthode consistant à créer un plan d'actions commun partagé par tous les agents lors de la réalisation de la tâche commune, les intentions conjointes [Jennings, 1992], qui est une technique reposant sur l'engagement de la part des agents coopérants envers des croyances et des buts mutuels, la reconnaissance de plans [Carberry, 2001], qui permet à un agent de reconnaître les intentions d'un autre en observant son comportement, et la planification par initiative mixte [Rich et al, 2000], qui est une technique de planification d'actions permettant à un usager et à un agent de travailler de façon parallèle, en donnant les mêmes possibilités aux deux d'intervenir durant les différentes étapes de réalisation de la tâche conjointe. Cette revue littéraire nous a fait prendre conscience de la problématique concernant la cohabitation entre l'utilisateur et son agent, ainsi que du besoin évident d'un modèle théorique de comparaison des points de vue comme celui présenté dans ce mémoire. Suite à cette recherche, nous avons étudié les différentes avenues possibles afin de développer une approche théorique répondant à la problématique identifiée. Cette étude nous a fait constater que l'ensemble de ces méthodes de coopération utilisent la théorie de l'interaction rationnelle, basée sur la logique des mondes possibles et ses extensions [Jennings, 1993a] [Wooldridge, 2000], pour faire communiquer un état mental à travers un acte de communication décrit par un langage tel que KQML [Finin et al., 1994]. De plus, ces méthodes ne prennent pas en considération la problématique citée précédemment. Par ailleurs, pour faciliter le raisonnement mutuel et minimiser cette

communication, nous avons plutôt eu l'idée d'utiliser la logique terminologique afin d'effectuer une comparaison des points de vue avant la prise de contrôle d'une tâche. Dans une seconde investigation, nous avons donc réalisé une recherche approfondie sur cette logique afin de connaître ses capacités et ses limites. Nous y avons découvert que l'utilisation de la logique terminologique comme support pour la comparaison des points de vue permettait de garantir leur égalité sémantique. L'originalité de notre approche se situe au niveau de l'utilisation de cette logique qui permet de ramener la problématique de la divergence terminologique à un problème de classification des concepts d'un point de vue dans un autre point de vue. En définitive, notre proposition consiste à utiliser la logique terminologique afin de créer une ontologie qui représente les points de vue de l'utilisateur et de l'agent, permettant ainsi de les comparer à l'aide d'heuristiques et en extraire une mesure de leur similarité. Cette mesure pourra ensuite servir à déterminer si une initiative mixte est sécuritaire, avant la prise de contrôle de la tâche commune.

Le contenu de ce mémoire est organisé en quatre chapitres.

Dans un premier temps, le chapitre 2 constituera un état de l'art sur la coopération dans une société d'agents. Il introduira d'abord les notions de base du domaine pour ensuite faire une revue des différentes méthodes de coopération citées précédemment. Ce chapitre permettra également de positionner notre travail dans ce domaine de la coopération multi-agents.

En deuxième lieu, le chapitre 3 se voudra une introduction détaillée à la logique terminologique et aux systèmes de représentation des connaissances terminologiques. Il sera, entre autres, question des origines de celle-ci, des concepts fondamentaux qui la compose, des différents formalismes existants ainsi que ses avantages et ses limitations. L'argument justifiant l'introduction détaillée de cette logique est qu'à l'heure actuelle, il n'existe aucun livre ou document grand public décrivant la logique terminologique, seulement quelques regroupements d'articles.

Enfin, le chapitre 4, qui constitue notre contribution dans le domaine de la coopération multi-agents, permettra de présenter le modèle formel de comparaison des points de vue. Dans un premier temps, nous présenterons les concepts de base de l'approche proposée, suivi d'une définition formelle du modèle. Finalement, un cas concret servant de validation à notre modèle et qui concerne le commerce électronique sera présenté.

La conclusion générale de ce mémoire permettra de présenter le bilan de notre contribution à l'avancement du domaine ainsi que les nouvelles voies de recherches intéressantes qui pourraient en découler.

CHAPITRE 2

COOPÉRATION DANS UN SYSTÈME MULTI-AGENTS

2.1 Introduction

L'évolution des systèmes d'intelligence artificielle a été très rapide durant les dernières années. De plus en plus, on s'intéresse non seulement à la création de systèmes intelligents capables d'inférer, mais également aux notions d'autonomie, de mobilité, de représentation de la connaissance, de résolution distribuée de problèmes, de communication, etc. C'est pour répondre à ces problématiques que depuis plusieurs années déjà, la communauté oeuvrant dans le domaine de l'intelligence artificielle distribuée [Jennings, 1993a] [Ferber, 1995] [Tambe, 1997] [Grosz et Kraus, 1998] [Wooldridge, 2000] tente d'élaborer des théories sur les concepts d'agent et de système multi-agents.

Le présent chapitre se veut un état de l'art entourant les systèmes multi-agents. Dans un premier temps, nous clarifierons les notions d'agent et de système multi-agents d'une façon précise. Ensuite, nous ferons une présentation des différentes méthodes de coopération qui peuvent être implantées dans une société d'agents. Dans cette partie, il sera question des approches et des mécanismes élémentaires permettant aux agents d'interagir, de communiquer et de coopérer entre eux d'une façon structurée et efficace. Finalement, la dernière partie du chapitre se voudra une présentation détaillée des récents travaux et des nouveaux développements dans le domaine des agents cognitifs coopérants.

2.2 La notion d'agent

Depuis les vingt-cinq dernières années, la communauté travaillant dans le domaine de l'intelligence artificielle pour les systèmes distribués essaie de définir de façon formelle

la notion d'agent. Jusqu'à maintenant, aucun consensus concernant une définition générale n'a été obtenu. Cependant, on peut constater que certaines définitions sont couramment utilisées dans la littérature. C'est notamment le cas de la définition de Wooldridge et Jennings [Wooldridge et Jennings, 1995], qui est l'une des plus reconnues et qui celle adoptée dans ce projet de recherche. Selon eux, un agent peut être défini comme étant une entité logicielle (virtuelle, module informatique) ou matérielle (robot) qui possède les quatre propriétés suivantes :

- **Autonomie :** cette propriété stipule qu'un agent devra fonctionner sans l'intervention directe d'un opérateur humain ou d'une autre entité. Elle définit également que cet agent aura un certain contrôle sur ses actions et sur son état interne.

- **Habileté sociale :** cette propriété définit qu'un agent devra interagir avec d'autres agents (ou possiblement des humains) via un langage de communication. Par exemple, un des langages utilisés est KQML (Knowledge Query and Manipulation Language), qui est un protocole de haut niveau basé sur les actes du langage [Finin et al., 1994].

- **Réaction :** cette propriété stipule qu'un agent devra percevoir son environnement et réagir en fonction des changements qui s'y produisent.

- **Pro-action :** cette propriété définit qu'un agent n'agira pas seulement en fonction de ce qui se passe dans son environnement, mais qu'il pourra agir et prendre des initiatives en fonction des buts qu'il désire atteindre.

Les agents sont *capables d'agir* et non pas seulement de raisonner comme dans les systèmes d'IA classique. L'action, qui est un concept fondamental pour les systèmes multi-agents, repose sur le fait que les agents accomplissent des actions qui vont modifier leurs environnements, y compris leurs états mentaux.

D'une façon plus formelle, on peut définir un agent A de manière minimale par le triplet suivant :

$$A = < \Sigma, O_p, EM >$$

Σ : représente l'ensemble des états possibles de l'environnement dans lequel évolue l'agent A

O_p : représente l'ensemble des opérations (actions) pouvant être exécutées par A

EM : représente l'ensemble des configurations possibles pour l'état mental de l'agent A

D'une manière générale, on peut alors voir un agent A comme étant une fonction f_A définie de la façon suivante :

$$f_A : \Sigma \times EM \rightarrow O_p$$

2.3 L'état mental d'un agent

Un agent artificiel intelligent, tout comme un être humain, possède un certain état d'esprit, un point de vue sur une situation. Cet état mental peut se définir comme étant la configuration dynamique de l'état interne d'un agent à un instant *t* [Rao et Georgeff, 1991]. Il est constitué des informations provenant des perceptions de l'environnement, aussi appelées les percepts, des objectifs de l'agent, des croyances de l'agent par rapport aux percepts et des intentions de celui-ci. Le modèle de référence pour la représentation de l'état mental d'un agent est le modèle BDI (Belief-Desire-Intention) développé par Rao et Georgeff [Rao et Georgeff, 1991] [Wooldridge et Jennings, 1994] [Rao et Georgeff, 1995].

- **Les croyances (Beliefs) :** les croyances sont le résultat des informations perçues de l'environnement et des connaissances inférées par l'agent à partir de ces informations. La figure 2.1.a nous montre un exemple de représentation des croyances d'un agent. Un agent A pourrait obtenir une connaissance P_1 de l'existence d'un état σ_1, provenant d'une perception $Percept_A(\sigma_1)$[1]. Il pourrait ensuite inférer une connaissance P_2, en effectuant une prévision $Prévision_A(P_1, R, C)$[2], à partir d'une série de règles d'inférence R et des croyances actuelles C, ce qui se traduirait de façon formelle par:

$$((P_1 = Percept_A(\sigma_1)) \wedge (P_2 = Prévision_A(P_1, R, C))) \Rightarrow (Bel_A P_1 \wedge Bel_A P_2)$$

[1] *$Percept_A(\sigma)$ est un opérateur (méthode) de perception indiquant que l'agent A perçoit une situation σ.*
[2] *$Prévision_A(P, R, C)$ est un opérateur (méthode) de prévision indiquant que l'agent A prévoit une situation future à partir d'une connaissance P, selon un certain nombre de règles R et selon ses croyances actuelles C.*

- **Les désirs (Desires) :** les désirs ou les buts d'un agent constituent sa motivation à agir [Wooldridge et Jennings, 1995]. Ces buts sont représentés par une série d'états que l'agent souhaite atteindre, tel que nous le montre l'exemple de la figure 2.1.b. Certains buts peuvent être implantés lors de la création de l'agent. Par exemple, un robot extracteur de minerai pourrait avoir comme objectif de base de ramener du minerai. D'autres désirs peuvent être inférés à partir des croyances de l'agent, par exemple lorsqu'un robot extracteur croit manquer de carburant, il pourrait inférer un nouvel objectif consistant à s'en procurer.

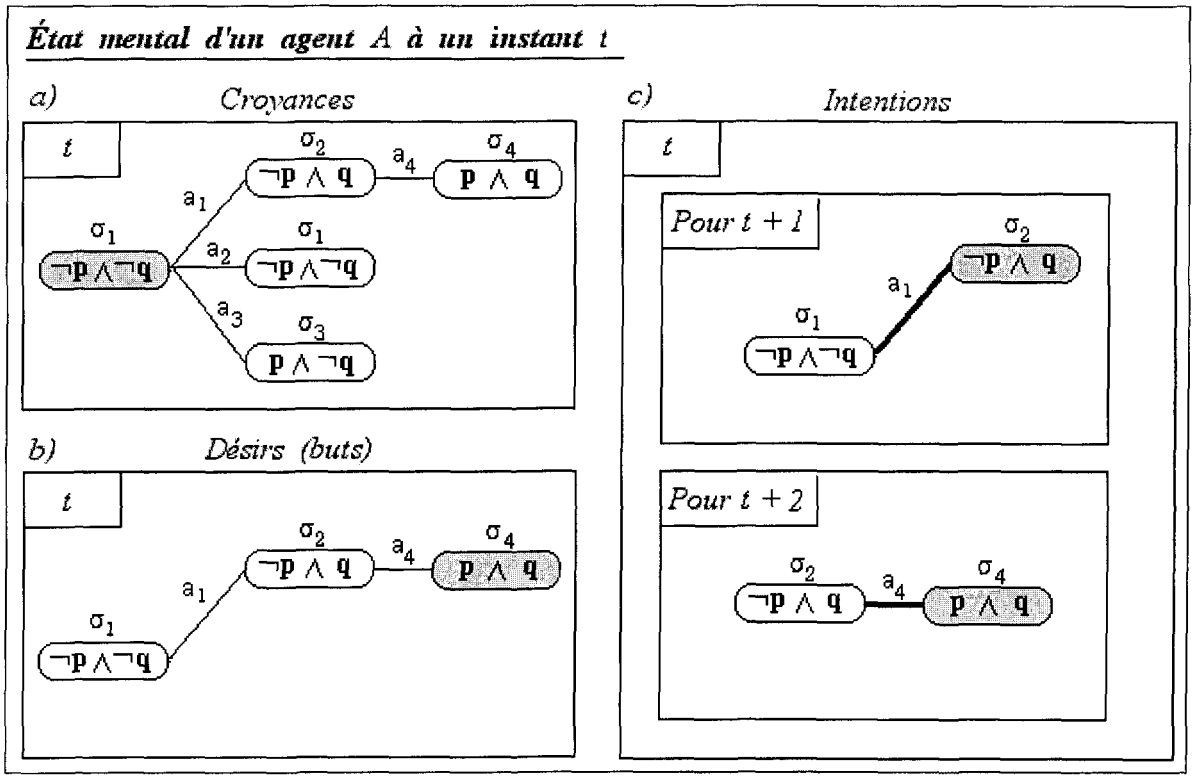

Fig. 2.1 : État mental d'un agent *A* à un instant *t*

- **Les intentions :** les intentions d'un agent représentent les actions que celui-ci compte effectuer dans le futur afin d'atteindre certains ou tous ses objectifs (désirs). La figure 2.1.c nous montre d'ailleurs un exemple de représentation des intentions d'un agent A à un instant t. Ces intentions, qui constituent en fait le plan d'actions de l'agent, sont inférées à partir des croyances et des buts de celui-ci. Par exemple, un robot extracteur qui a pour objectif de ramener du minerai pourrait inférer de commencer à creuser la roche dans 10 minutes, s'il croit qu'il arrivera près du gisement seulement dans 10 minutes.

La figure 2.1 illustre bien le fonctionnement du modèle BDI. On peut y voir qu'à un instant t, l'agent A croit en un état actuel σ_1 de l'environnement. Il croit également qu'il est possible pour lui d'effectuer les actions a_1, a_2 et a_3 à l'instant $t+1$ permettant ainsi d'atteindre respectivement les états σ_1, σ_2, σ_3. On peut voir que l'agent A a pour objectif d'atteindre l'état σ_4 et que pour cela, il lui faut préalablement atteindre l'état σ_2. Finalement, on voit que les intentions de l'agent A à l'instant t sont d'effectuer l'action a_1 à l'instant $t+1$ afin d'atteindre l'état σ_2, et ensuite d'exécuter l'action a_4 à l'instant $t+2$, ce qui lui permettra d'arriver à son but qui est l'état σ_4.

2.4 Les systèmes multi-agents (SMA)

Un système multi-agents, tel qu'illustré à la figure 2.2, est composé des éléments suivants [Ferber, 1995] :

a) Un environnement E, c'est-à-dire un espace disposant généralement d'une métrique.

b) Un ensemble d'objets O. Ces objets sont situés, c'est-à-dire que pour tout objet, il est possible, à un moment donné, d'associer une position dans E. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.

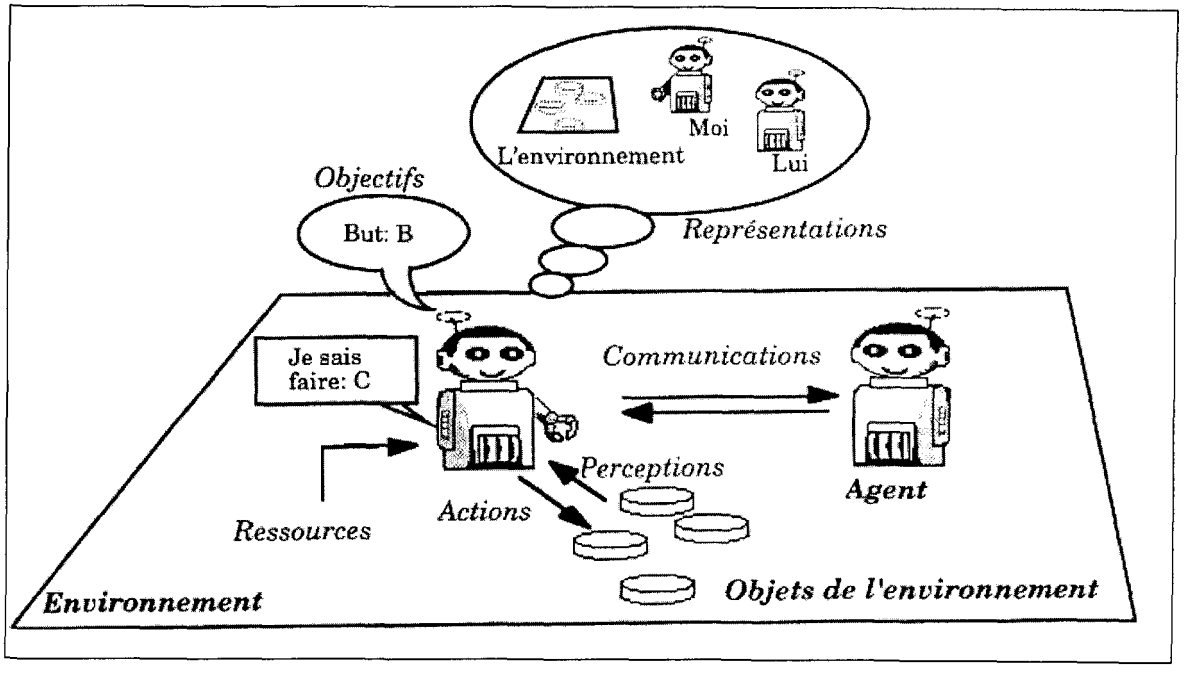

Fig. 2.2 : Représentation d'un système multi-agents, d'après Ferber [Ferber, 1995]

c) Un ensemble d'agents A, qui sont des objets particuliers ($A \subseteq O$), lesquels représentent les entités actives du système.

d) Un ensemble de relations R qui unissent des objets (et donc des agents) entre eux.

e) Un ensemble d'opérations O_p permettant aux agents de A de percevoir, produire, consommer, transformer et manipuler des objets de O.

f) Un ensemble d'opérateurs permettant d'exécuter les différentes opérations O_p et de produire la réaction du monde à ces opérations. C'est ce qu'on appelle les lois de l'univers.

Il existe certains cas particuliers de systèmes dans lesquels $A = O$ et $E = \varnothing$. Dans ce cas, les relations R définissent un réseau : chaque agent est lié directement à un ensemble d'autres agents. Ces systèmes, qui sont appelés des SMA *purement communicants*, sont très souvent utilisés dans le domaine de l'intelligence artificielle distribuée. Leur domaine de prédilection est la coopération de modules logiciels dont la fonction est de résoudre un problème ou d'élaborer une expertise (par exemple, l'interprétation de signaux), à partir de modules spécialisés, comme dans le cas d'un système de contrôle distribué, où E est défini par la structure du réseau sous-jacent. Ces systèmes se caractérisent par le fait que les interactions soient essentiellement des communications intentionnelles et que leur mode de travail ressemble à celui d'un organisme social (groupe de travail, entreprise, administration, etc).

2.5 L'interaction dans un SMA

La notion d'interaction est au centre de la problématique entourant les systèmes multi-agents [Demazeau et Müller, 1991]. Une interaction est une mise en relation

dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Ces interactions permettent aux agents d'augmenter leur puissance, d'atteindre leurs buts, de communiquer leur savoir, d'augmenter la vitesse avec laquelle ils exécuteront une tâche, etc. Les interactions entre agents sont également à l'origine de problèmes de conflits causés par des divergences d'objectifs, d'opinions, bref, par le fait qu'ils interagissent entre eux.

2.5.1 Les situations d'interaction

Il existe une multitude de situations pouvant donner l'opportunité aux agents d'interagir entre eux. Il est possible de classifier ces différentes situations par rapport à trois principaux critères : les objectifs des agents, les relations que les agents entretiennent envers les ressources qu'ils possèdent et les moyens (ou compétences) dont ils disposent pour parvenir à leurs fins [Ferber, 1995]. Ces critères nous permettent de faire une typologie des situations d'interaction, comme nous le montre le tableau 2.1. On peut voir que lorsque les buts des agents sont compatibles, il est possible et même dans plusieurs cas souhaitable pour les agents de coopérer les uns avec les autres. Cependant, lorsque leurs buts sont incompatibles, une multitude de conflits émergent.

Buts	Ressources	Compétences	Types de situation	Catégorie
Compatibles	Suffisantes	Suffisantes	Indépendance	Indifférence
Compatibles	Suffisantes	Insuffisantes	Collaboration simple	Coopération
Compatibles	Insuffisantes	Suffisantes	Encombrement	
Compatibles	Insuffisantes	Insuffisantes	Collaboration coordonnée	
Incompatibles	Suffisantes	Suffisantes	Compétition individuelle	Antagonisme
Incompatibles	Suffisantes	Insuffisantes	Compétition collective	
Incompatibles	Insuffisantes	Suffisantes	Conflits individuels	
Incompatibles	Insuffisantes	Insuffisantes	Conflits collectifs	

Tableau 2.1 : Classification des situations d'interaction

- **Indépendance :** ce type de situation ne pose aucun problème du point de vue multi-agents et se résume en une simple juxtaposition des actions des agents pris indépendamment, sans qu'il y ait effectivement d'interactions.

- **Collaboration simple (délégation):** ce type de situation consiste en une simple addition des compétences qui ne nécessite aucune coordination entre les intervenants. Par exemple, un agent qui ne possède pas les compétences pour effectuer une tâche pourrait tout simplement la faire exécuter par un autre agent [Castelfranchi et Falcone, 1998].

- **Encombrement :** ce type de situation entraîne que les agents se gênent mutuellement dans l'accomplissement de leurs tâches alors qu'ils n'ont pas besoin les uns des autres. Par exemple, il peut survenir des problèmes d'encombrement d'un réseau lorsque les agents distants effectuent un trop grand nombre de communications.

- **Collaboration coordonnée :** ce type de situation implique que les agents devront coordonner leurs actions afin d'arriver à atteindre leurs objectifs [Ciancarini et al., 1999]. Ne disposant ni des ressources, ni des compétences nécessaires, les agents doivent coopérer. Par exemple, un robot extracteur de minerai pourrait coopérer avec un robot spécialisé dans la recherche de minerai. Le premier robot pourrait coordonner ses actions et extraire le minerai seulement une fois que le second aurait identifié un gisement.

- **Compétition individuelle :** ce type de situation survient lorsque les agents ont des objectifs incompatibles. En d'autres termes, la réalisation des objectifs de l'un implique que les autres ne pourront pas réaliser les leurs. Par exemple, si l'objectif de deux agents consiste à gagner une partie d'échecs qu'ils disputent ensemble, la réalisation du but de l'un entraînera nécessairement l'échec de l'autre.

- **Compétition collective :** ce type de situation survient lorsque des agents qui ont des buts incompatibles ne possèdent pas les compétences nécessaires pour réaliser leurs objectifs. Dans ce cas, les agents doivent se regrouper au sein de coalitions afin de parvenir à leurs buts. Un exemple caractéristique est la compétition en équipe, comme dans un match de soccer [Spaan, 2002].

- **Conflits individuels pour des ressources :** ce type de situation implique que les ressources des agents sont insuffisantes et ne peuvent être partagées. L'accès aux ressources devient donc une source de conflits, où chaque agent tente d'acquérir suffisamment de ressources pour réaliser ses buts; par exemple, plusieurs programmes en concurrence pour l'utilisation d'un périphérique ou d'une imprimante à un moment donné.

- **Conflits collectifs pour des ressources :** ce type de situation combine la compétition collective aux conflits individuels pour des ressources. Par exemple, on peut penser à plusieurs robots jouant une partie de soccer qui sont en compétition

collective pour l'obtention de la seule et unique ressource, la balle (Robocup'97) [Tambe et al., 1997].

2.5.2 La coopération

La coopération se définit comme étant la mise en commun de ressources et de compétences par un groupe d'agents afin d'arriver à un but commun, et/ou à l'augmentation de leurs performances mutuelles [Galliers, 1991] [Doran et al., 1997]. Par exemple, dans le cas de robots extracteurs de minerai, on pourrait penser que leur regroupement résulterait en une sommation simple de leurs performances individuelles. Autrement dit, si un robot rapporte 10 kilos de minerai par heure, 5 robots travaillant de concert rapporteront 50 kilos de minerai par heure. Cependant, l'amélioration de leurs performances pourrait être beaucoup plus que linéaire; par exemple, si certains robots se concentrent à creuser le minerai et que les autres s'activent à le ramener, l'efficacité individuelle des robots pourrait passer de 10 à 12 kilos à l'heure. Afin de pouvoir observer cette augmentation, il faudrait quantifier l'amélioration de performances à l'aide d'un indice. Dans le cas présent, cet indice d'amélioration pourrait se calculer de la façon suivante :

$$\sigma_{amélioration} = P_{total} - ((P_{ind} * n) + a)$$

P_{ind} : *performance individuelle d'un agent*

n : *nombre d'agents*

a : *performances perdues pour la gestion de la coopération*

P_{total} : *performance brute totale des robots*

Il existe deux grandes écoles de pensée concernant la coopération :

a) **La coopération comme attitude intentionnelle :** cette école de pensée définit que la coopération résulte d'une attitude qu'ont les agents à vouloir coopérer après avoir identifié un but commun [Conte et al., 1991] [Galliers, 1991]. Par exemple, deux agents qui auraient besoin du résultat d'un calcul mathématique complexe pourraient décider de s'associer, afin que chacun effectue la moitié de la tâche de calcul pour ensuite partager le résultat. Cette décision intentionnelle de la part des agents constituerait donc la preuve d'une véritable volonté de coopération.

b) **La coopération du point de vue de l'observateur :** cette école de pensée considère la coopération comme une qualification de l'activité d'un ensemble d'agents par un observateur extérieur qui n'aurait pas accès à l'état mental des agents. Par exemple, si l'on peut qualifier le comportement des fourmis de coopératif, c'est parce que, en tant qu'observateur, on dénote un certain nombre de phénomènes que l'on utilise comme des indices d'une activité de coopération. Il est donc possible de définir un ensemble d'indices observables (partage de ressources, parallélisation d'actions, etc.) permettant de qualifier une situation comme étant de la coopération [Durfee et al., 1989] [Bouron, 1992].

L'énigme de la coopération se résume par la formule suivante [Ferber, 1995] :

Coopération = collaboration + coordination d'actions + résolution des conflits

2.5.2.1 La collaboration

Cette technique consiste à faire travailler plusieurs agents sur un projet en répartissant des tâches, des informations et des ressources de manière à réaliser une œuvre commune - technique de partage de tâches - [Castelfranchi et Falcone, 1998]. La solution utilisée pour la répartition de tâches passe par l'utilisation d'un agent coordonnateur centralisé [Tambe, 1997][Tuomas, 1997]. Cet agent se verra donc attribuer le rôle de distribuer les tâches aux agents disponibles, selon leurs compétences et les besoins. Dans ce type de système, deux approches s'opposent :

a) **Réseau d'accointances:** cette approche consiste à répartir les tâches selon les compétences des agents et les disponibilités de ceux-ci. Pour ce faire, l'agent centralisé utilise un modèle de représentation mutuelle des capacités de chacun, que l'on appelle un réseau d'accointances [Gasser et al., 1987].

b) **Technique d'appel d'offre :** cette méthode repose sur le modèle de l'offre et de la demande qu'utilisent les entreprises. Pour cela, un agent coordonnateur diffusera une tâche à tous les agents. Les agents qui le désirent pourront alors faire une offre de service pour ce travail. L'une des

utilisations les plus connues de cette approche est le réseau contractuel développé par Davis [Davis et Smith, 1983].

2.5.2.2 La coordination d'actions

Cette technique suppose que la gestion d'un ensemble d'agents génère un certain nombre de tâches supplémentaires, appelées tâches de coordination, qui ne sont pas directement productives, mais qui servent simplement à faire en sorte que les actions productives puissent s'accomplir dans les meilleures conditions [Durfee et Lesser, 1991]. Par exemple, un robot transporteur ne peut pas ramener du minerai s'il n'a pas préalablement été extrait. Un modèle formel pour l'implantation d'un système de coordination a été proposé par Ciancarini [Ciancarini et al., 1999]. La coordination des différentes tâches d'un groupe d'agents peut s'effectuer grâce à un agent coordonnateur (comme c'est le cas pour la répartition de tâches) ou s'organiser directement entre deux ou plusieurs agents. D'autres méthodes récentes ont également été suggérées, comme les plans partagés [Crosz et Kraus, 1998], les intentions conjointes [Jennings, 1992] et la délégation et l'adoption [Haddadi, 1996]. Ces méthodes seront d'ailleurs décrites dans la dernière section du chapitre.

2.5.2.3 La résolution de conflits

Une méthode de coopération s'appuie sur le fait que des agents qui cohabitent ensemble seront probablement confrontés à des situations conflictuelles. La résolution de ces conflits peut donc être considérée comme une méthode de coopération [Tessier et al.,

2001]. L'une des méthodes utilisées est la négociation par argumentation [Kraus et al., 1998], qui sera décrite en détail dans la dernière section du chapitre.

2.6 Les conflits entre les agents dans un SMA

2.6.1 La notion de conflit

La vie en société, que ce soit une société humaine ou une société d'agents, implique l'apparition de conflits. Par ailleurs, on peut dénoter certaines conditions qui sont préalables à l'émergence d'un conflit [Tessier et al., 2001]:

- **Différence entre des éléments communs :** cette condition signifie qu'un conflit entre des agents ne pourra apparaître que si certaines différences existent en relation avec les éléments qu'ils ont en commun [Chaudron et al., 2000]. Par exemple, des conflits apparaissent lorsque plusieurs agents veulent utiliser les mêmes ressources en même temps ou lorsqu'il y a apparition d'une contradiction entre les différentes croyances, intentions ou buts des agents.

- **Au moins deux agents :** cette condition définit qu'à un instant t, aucune différence n'existe entre un agent A et lui-même. Un agent ne peut donc pas croire en la véracité d'une réalité x et ne pas y croire en même temps. Par ailleurs, cette affirmation peut sembler contradictoire avec la définition des « conflits internes » de Castelfranchi [Castelfranchi, 2000], qui stipule qu'un agent peut entretenir des conflits internes en rapport avec les éléments qu'il possède. Cependant, dans le cas

où un agent possède deux points de vue contradictoires d'une même situation, on peut alors considérer chacun des points de vue de celui-ci comme étant un agent distinct qui ne possède aucun conflit interne.

- **Symétrie d'un conflit:** cette condition stipule que si un agent *A* est en situation de conflit avec un agent *B*, alors l'agent *B* est automatiquement en situation de conflit avec l'agent *A*.

- **Contexte commun :** cette condition signifie que pour qu'il existe des conflits entre deux agents, ils doivent avoir des éléments en commun, que ce soit des ressources, des règles, des niveaux de décision, etc. Par exemple, même si la divergence des croyances est une condition nécessaire à l'apparition de conflits [Castelfranchi, 2000], ces croyances doivent tout de même appartenir au même contexte et être fondées sur des bases communes afin de pouvoir être comparées.

Il existe deux grandes familles de conflits : les conflits physiques et les conflits de connaissances [Tessier et al., 2001].

2.6.2 Conflits physiques

Ce type de conflit survient au niveau des ressources, on peut l'appeler conflit extrinsèque ou conflit non analytique [Castelfranchi, 2000]. Les objectifs des agents ne sont pas logiquement contradictoires ou divergents, mais ils deviennent incompatibles parce qu'il n'y a pas suffisamment de ressources pour que tous les agents atteignent leurs buts.

L'exemple typique est l'espace physique partagé par deux robots. Ces deux robots ne peuvent être au même endroit au même moment. Une définition formelle des conflits d'espace entre des robots est proposée par Penders [Penders, 1999]. Dans la plupart des systèmes multi-robots, les conflits d'espace sont courants et des stratégies de planification des tâches doivent être instaurées pour les éviter.

2.6.3 Conflits de connaissances

Ce type de conflit se situe au niveau de la connaissance. Il survient lorsque les croyances, les connaissances ou les plans d'actions des agents sont contradictoires ou divergents. Il peut aussi être appelé conflit intrinsèque ou conflit analytique. Par exemple, si un agent A croit en une réalité p et qu'un agent B nie cette même réalité, ils sont en contradiction. Il n'y a pas vraiment d'ambiguïté possible dans ce genre de situation. Les contradictions, qu'elles soient directes ou indirectes, peuvent être exprimées de façon formelle, à l'aide de « croyances nécessaires » pour réaliser un but [Castelfranchi, 2000] :

Contradiction directe : $(Goal\ g\ /\ Bel_A\ p) \wedge (Goal\ g\ /\ Bel_A\ \neg p)$

Contradiction indirecte : $(Goal\ g\ /\ Bel_A\ p) \wedge (Goal\ g\ /\ Bel_A\ q) \wedge (Bel_A\ (p \rightarrow \neg q))$

Goal : *désigne un opérateur modal de buts du modèle BDI* [Rao et Georgeff, 1991]

Bel : *désigne un opérateur modal de croyances du même modèle BDI*

Dans cet exemple, on voit qu'une contradiction directe (un conflit interne) survient lorsqu'un agent A ayant un certain but g, croit en même temps en la véracité d'un fait p et en son inverse. Dans le cas d'une contradiction indirecte, elle survient lorsqu'un agent A, ayant un certain but g, croit aux réalités p et q, tout en étant convaincu que la véracité de p entraîne la négation de q. D'autre part, il est également possible de définir les contradictions entre les points de vue de deux agents A et B, ainsi que les contradictions internes du point de vue d'un agent, à l'aide d'une série de formules bien formées qui les caractérisent [Tessier et Chaudron, 1998] :

$\phi \wedge \neg \phi$	:	*l'action ϕ est considérée réalisée et non réalisée au même moment*
$Bel_A\phi \wedge Bel_B\neg \phi$	:	*l'agent A croit en la réalisation de ϕ et B ne croit pas en ϕ*
$Bel_A\phi \wedge Bel_A(Bel_B\neg \phi)$	:	*l'agent A croit en la réalisation de ϕ et croit que B ne croit pas en ϕ.*
$Wants^3{}_A\phi \wedge Wants_B\neg \phi$	:	*l'agent A désire « Wants » la réalisation de ϕ et B ne la désire pas*
$O_A\phi \wedge F_B\neg \phi$	:	*l'agent A a l'obligation « O » d'effectuer l'action ϕ et B a l'interdiction « F » d'exécuter ϕ*
$Does_A\phi \wedge F_A\ \phi$	:	*l'agent A effectue l'action ϕ alors que celle-ci lui est interdite*

[3] *Wants*, *O*, *F*, *Does* et *CannotDo* sont des opérateurs modaux de l'extension du modèle BDI.

$O_A\phi \wedge CannotDo_A\neg\phi$: *l'agent A a l'obligation d'exécuter l'action ϕ sachant qu'il n'en a pas la capacité*

2.7 Les récents développements dans les méthodes de coopération

2.7.1 Les plans partagés

Dans les dernières années, plusieurs modèles ont été développés afin de fournir un cadre pour l'implémentation de la coopération à travers un groupe d'agents. L'un des modèles les plus prometteurs, appelé le modèle par *plans partagés*, a été développé par Crosz et Kraus [Crosz et Kraus, 1993], [Crosz et Kraus, 1996], [Crosz et Kraus, 1998]. Ce modèle est basé sur le fait que dans un contexte réel, l'action conjointe d'une équipe ne consiste pas simplement en une série d'actions individuelles exécutées en parallèle ou de façon hiérarchisée, mais plutôt que chaque membre du groupe doit connaître l'état actuel de la réalisation de la tâche commune et être concerné par sa bonne marche. Crosz et Kraus ont donc suggéré de prendre en compte cet aspect afin de créer un modèle viable de coopération. Leur proposition consiste en la création d'un plan commun, partagé par tous les agents devant coopérer ensemble. Selon ce modèle, le plan sera créé grâce à la mise en commun de tous les plans individuels (intentions) des agents du groupe. La conception de ce plan est donc effectuée en tenant compte des connaissances de tous les agents. Ensuite, chacun des agents devra réaliser une certaine partie du plan conjoint.

On peut définir le plan individuel d'un agent A comme étant une série d'actions permettant de réaliser un acte α [Hadad et Kraus, 1999]. Afin que son plan soit complet,

l'agent A doit également satisfaire les quatre conditions suivantes : (1) A doit savoir comment exécuter α, c'est-à-dire qu'il doit connaître toutes les sous-actions nécessaires à sa réalisation, (2) A doit croire qu'il est apte à réaliser toutes les sous-actions de α, (3) A doit avoir l'intention d'exécuter toutes les sous-actions de α, (4) A doit avoir un plan d'actions individuel pour l'exécution de chaque sous-action de α qui ne constitue pas une action atomique. Un groupe d'agents qui désire créer un plan partagé doit d'abord croire en la nécessité de réaliser cet acte α. Le plan partagé entre ces agents peut être défini comme étant la somme des actions identifiées comme étant nécessaires par l'ensemble des agents afin d'exécuter un acte α. Ce plan d'actions partagé représentera les croyances mutuelles des agents.

Pour illustrer le mécanisme général du modèle, nous allons prendre un exemple tiré du commerce électronique [Hadad et Kraus, 1999]. Supposons d'abord l'existence de deux entreprises possédant chacune un agent vendeur et un agent acheteur. L'agent acheteur a pour mission de se procurer les produits nécessaires au bon fonctionnement de son entreprise et l'agent vendeur doit vendre les produits fabriqués par son entreprise à d'autres agents acheteurs. Dans le cas où l'agent acheteur de la première entreprise, noté A_1, déterminera qu'un produit nécessaire à l'entreprise est manquant, il devra alors trouver un vendeur. Admettons maintenant que A_1 ait décidé de faire affaires avec l'agent vendeur de la deuxième entreprise, noté V_2. Les deux agents devront s'entendre sur un plan d'actions partagé, à l'aide de leur plan individuel respectif, afin d'exécuter la transaction. Une fois que ce plan d'actions partagé sera déterminé, le plan individuel de chacun des agents sera

modifié afin de prendre en compte ce nouveau plan. Chacun des agents se verra attribuer la tâche d'exécution d'une partie du plan partagé. La figure 2.3 illustre cette situation:

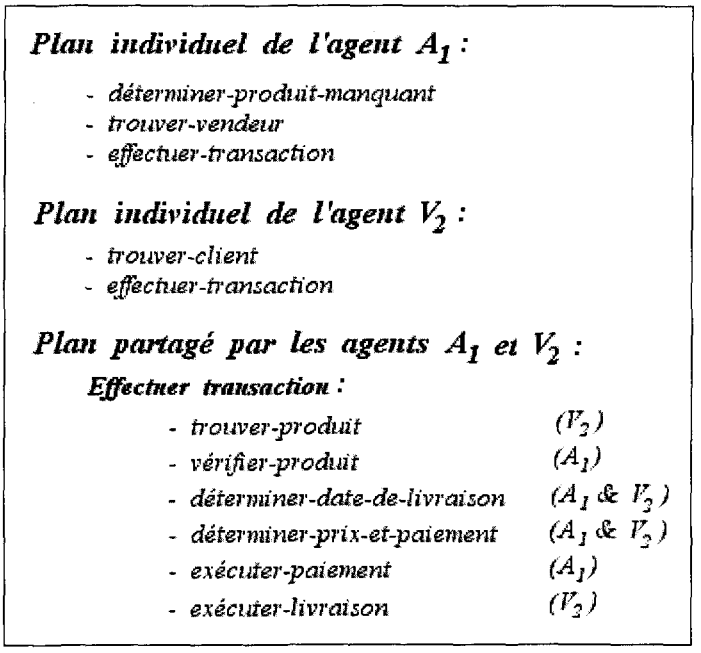

Fig. 2.3 : Plan de réalisation conjoint d'une transaction

Les différentes parties du plan d'actions partagé seront distribuées entre les deux agents de la façon suivante : d'abord l'agent V_2 trouvera le produit demandé dans son stock en exécutant l'action *trouver-produit*, ensuite l'agent A_1 s'assurera de la conformité de ce produit avec l'action *vérifier-produit*, les deux agents effectueront ensuite conjointement les actions *déterminer-date-de-livraison* et *déterminer-prix-et-paiement*, l'agent A_1 effectuera son paiement avec l'action *exécuter-paiement* et finalement, l'agent V_2 se chargera de la livraison avec *exécuter-livraison*. On peut également voir plusieurs autres exemples d'utilisation du modèle de Crosz et Kraus dans divers domaines; par exemple, on

peut noter son utilisation dans un système de jeu de soccer robotisé [Spaan, 2002], dans le développement d'interfaces homme-machine [Rich, 2000] ainsi que dans l'interprétation de conversations en langage naturel [Lochbaum, 1994].

Les avantages d'un tel modèle sont nombreux. En plus de travailler en fonction d'un but conjoint, tous les agents auront une idée commune sur la façon d'atteindre ce but, ce qui évitera plusieurs conflits. Ensuite, le plan commun est inféré à partir des connaissances de tous les agents, ce qui permet de maximiser et de valider celui-ci. Par ailleurs, il est très facile de modifier et de réajuster ce plan partagé en fonction des nouvelles données qui proviennent de chacun des agents. Finalement, le nombre de communications entre les agents sera réduit grâce à cette centralisation.

Bien que ce modèle fournisse une structure permettant l'implantation d'un système de coopération, il ne répond que d'une façon partielle à cette vaste problématique. Par ailleurs, son utilisation implique forcément que tous les agents du système utilisent la même notation pour décrire leurs plans d'actions, et qu'ils emploient les mêmes termes lorsqu'ils décrivent la même situation. Il est donc facilement possible d'implanter ce modèle sur un groupe d'agents développé de façon standard par un même consortium. Cependant, l'utilisation d'un tel modèle entre plusieurs agents développés avec des standards, des notations et des langages différents, est, a priori, impossible. Afin d'implanter ce modèle, tous les agents doivent pouvoir comparer leurs plans d'actions individuels avec celui des autres de manière à pouvoir créer un plan partagé.

2.7.2 Les intentions conjointes

Une intention conjointe constitue un engagement (*commitment*) par un groupe d'agents à effectuer un acte collectif en partageant un certain état mental commun [Cohen et Levesque, 1991] [Jennings, 1992] [Jennings, 1993a] [Jennings, 1993b]. Afin d'arriver à résoudre une tâche collectivement, Cohen et Levesque stipulent que chacun des agents du groupe devra s'engager à vouloir atteindre les objectifs collectifs. Cet engagement obligera chacun d'eux à conserver ces buts, tant et aussi longtemps que l'action commune durera. Ces objectifs, qui sont appelés des buts conjoints persistants, constituent l'état mental partagé par les agents. Pour cela, dans un premier temps, les agents doivent acquérir des croyances mutuelles concernant l'état du problème. Ces croyances constituent une sorte de base de connaissances conjointes permettant aux agents d'avoir une vision commune du problème. Par exemple, un agent A et un agent B pourraient croire tous les deux, de façon individuelle, en une réalité p. Afin de créer une croyance mutuelle en p, les agents A et B devront respectivement croire en p et également penser que leur homologue croit lui aussi en p. Ce qui se formalise de la façon suivante :

$$MB^{4}_{AB}\ p \equiv ((Bel_A\ p) \wedge (Bel_A\ (Bel_B\ p))) \wedge ((Bel_B\ p) \wedge (Bel_B\ (Bel_A\ p)))$$

Dans un deuxième temps, les agents devront s'engager à la réalisation d'un certain nombre d'objectifs communs. Ces objectifs seront persistants et ne pourront être abandonnés par un agent durant la collaboration. Encore une fois, chacun des agents devra

[4] MB désigne un opérateur modal proposé par la théorie des intentions conjointes qui désigne une croyance mutuelle entre deux agents

croire que les autres ont les mêmes objectifs que lui. Par exemple, un agent A et un agent B pourraient s'engager mutuellement à la réalisation d'un objectif conjoint g, ce qui se traduirait formellement de la façon suivante :

$$MG^{5}_{AB}\ g \equiv ((Goal_A\ g) \wedge (Bel_A\ (Goal_B\ g))) \wedge ((Goal_B\ g) \wedge (Bel_B\ (Goal_A\ g)))$$

Une fois ces buts conjoints déterminés, il faudra que les agents construisent une solution commune, qui correspondra à leurs intentions conjointes. La totalité de cette solution conjointe peut être représentée sous la forme d'une séquence logique permettant aux agents de connaître leur rôle ainsi que le moment où ils devront entrer en actions. Par exemple, deux robots A et B, chargés de l'extraction et du transport du minerai, dans une mine n'ayant de place que pour un seul à la fois, pourraient avoir un cycle d'interaction conjoint défini de la façon suivante :

$$((e_1, A)\ ;\ (e_2, A);\ (e_1, A));\ (e_2, A))\ \|\ ((e_2, B)\ ;\ (e_1, B);\ (e_2, B);\ (e_1, B))$$

e_1 = *Effectuer l'extraction du minerai*

e_2 = *Effectuer le transport du minerai*

Dans cet exemple, le symbole « ; » est utilisé pour signifier qu'une action est subséquente à une autre, et le symbole « || » désigne que les séquences à gauche et à droite de celui-ci s'effectueront en parallèle. Les intentions conjointes des deux robots de

---

[5] MG désigne un opérateur modal représentant un but conjoint entre deux agents

l'exemple ci-haut sont donc de coordonner leurs actions afin que, lorsque l'un soit en train d'extraire du minerai, l'autre effectue le transporte du sien. De cette façon, ils pourront tous deux extraire du minerai sans nuire à leurs activités respectives.

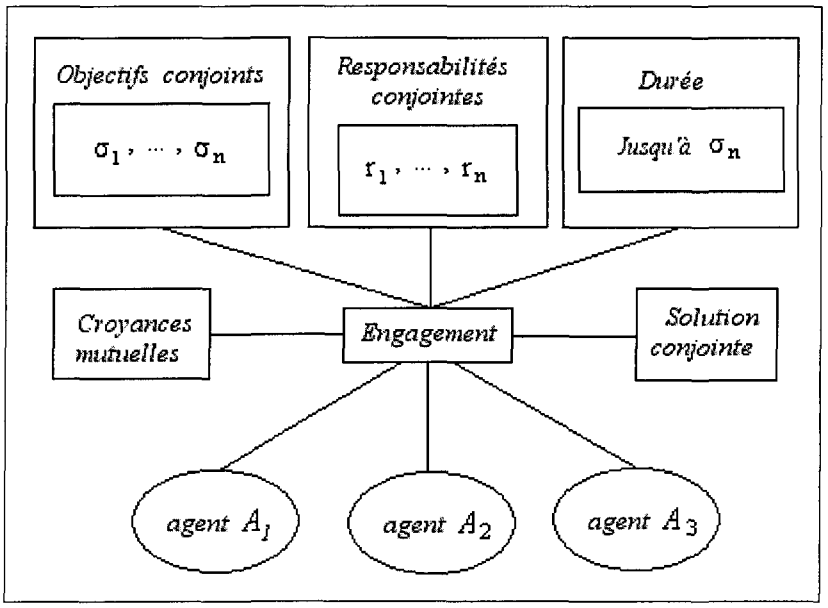

Fig. 2.4 : Modèle d'intentions conjointes

La théorie sur les intentions conjointes a été enrichie par Jennings [Jennings, 1993a], dans les années 90, en introduisant la notion de responsabilités conjointes. Cette notion signifie que les agents, en plus de s'engager à la réalisation d'un certain nombre d'objectifs communs, devront également s'engager à obéir à une série de règles de conduite en société, qui sont appelées les responsabilités conjointes des agents, tel qu'illustré à la figure 2.4. Ces responsabilités permettront d'améliorer la collaboration en fixant des bases communes et des règles strictes à respecter lors des actions de coopération. Par exemple, une règle de responsabilité pourrait obliger les agents coopérants à remettre leurs résultats

partiels de solutions en utilisant un format standardisé. La figure 2.4 nous montre un modèle d'intentions et de responsabilités conjointes composé de trois agents, A_1, A_2 et A_3. Avant de débuter leur collaboration, les trois agents s'engagent à vouloir atteindre un certain nombre d'objectifs conjoints correspondant à une série d'états $\{\sigma_1, \ldots, \sigma_n\}$, à s'acquitter d'une série de responsabilités $\{r_1, \ldots, r_n\}$ et cela, jusqu'à ce que l'état σ_n soit atteint.

L'approche par intentions conjointes est à la base de nombreux travaux sur la coopération [Searle, 1991] [Lochbaum et al., 1991] [Rao et al., 1992] [Jennings, 1993b]. Elle constitue un cadre pouvant facilement être adapté et intégré à plusieurs modèles de coopération. Cette approche peut notamment être utilisée en conjonction avec d'autres modèles populaires existants, comme le modèle de plans partagés [Crosz et Kraus, 1998] ou celui de la délégation [Castelfranchi et Falcone, 2000]. Par exemple, la solution de réalisation conjointe pourrait être implantée à l'aide d'un plan partagé.

Bien que cette approche fournisse un modèle de base permettant le développement de théories complexes sur la coopération, elle néglige certains aspects de cette problématique. Par exemple, toute la problématique entourant les conflits lors de la collaboration est quasi absente, seules les règles de responsabilités peuvent être utilisées afin de minimiser les conflits. Finalement, il ne faut pas oublier que la communication entre les agents lors de leurs engagements conjoints nécessite l'utilisation d'un même protocole

ou langage afin que ceux-ci puissent se comprendre. Il serait donc difficile d'utiliser cette approche dans un système multi-agents hétérogène.

2.7.3 La négociation par argumentation

La résolution de conflits est un problème critique dans les systèmes de coopération multi-agents [Tessier et al., 2001]. Plusieurs techniques de résolution ont été proposées, l'une des plus prometteuses est la *négociation via l'argumentation* (NVA) [Kraus et al., 1998]. Elle impose aux agents en conflit de fournir une justification explicite aux autres, à l'aide d'arguments, de leur plan d'actions. Une fois cette justification effectuée, chaque agent peut étudier les arguments des autres et réviser son plan d'actions en tenant compte de ces nouveaux arguments. Les agents en conflit peuvent également faire des propositions et des contre-propositions aux autres agents, toujours appuyées d'arguments pour les justifier.

Afin de mieux comprendre le fonctionnement de cette approche, nous allons prendre un exemple de scénario d'argumentation, tiré du système de simulation de combat en hélicoptère CONSA (Collaborative Negociation System based on Argumentation) développé par Tambe et Jung [Tambe et Jung, 1999]. Considérons d'abord deux agents pilotes, A_1 et A_2, et deux positions ennemies, E_1 et E_2, où A_1 reçoit les informations à propos de E_1 et A_2 reçoit celles concernant E_2. Supposons maintenant que A_1 et A_2 soient situés à 100 mètres l'un de l'autre, et qu'un conflit survienne dû au fait qu'il est normalement requis de conserver au moins 1 kilomètre de distance entre les deux. C'est à ce moment que

la négociation par argumentation aura lieu afin de déterminer la marche à suivre pour régler cette situation. L'agent A_1 calculera d'abord ses arguments pour la négociation et construira sa proposition de règlement pour le conflit. Ensuite, il communiquera cette suggestion de plan d'actions à A_2 *{(A_1 déplace 450m gauche, A_2 déplace 450m droite)}*, ainsi que ses justifications pour ce plan d'actions *{(ennemie position E_1, séparation = 100m)}*. Lorsque A_2 recevra la proposition, il évaluera que celle-ci est inacceptable, car il est impossible pour lui de se déplacer de 450 mètres vers la droite à cause de la position ennemie en E_2. L'agent A_2 déterminera ensuite que le maximum de déplacement possible pour lui vers la droite est de 300 mètres, il enverra donc une contre-proposition à A_1 *{(A_1 déplace 600m gauche, A_2 déplace 300m droite)}* ainsi que ses arguments *{(ennemie position E_1, ennemie position E_2, séparation = 100m)}*. Finalement, l'agent A_1 évaluera cette contre-proposition d'acceptable et le conflit sera réglé.

Bien que l'approche originale fut très prometteuse, on déplorait l'absence d'un modèle formel pour l'implantation de cette théorie de négociation. Cependant, cette lacune a été récemment comblée par les travaux de Jung [Jung et al., 2001], qui a proposé un modèle formel pour l'argumentation basé sur l'algorithme de Yokoo et Hirayama [Yokoo et Hirayama, 1998]. Cet algorithme a été développé à l'origine pour résoudre des problèmes de satisfaction de contraintes distribuées. Un problème de satisfaction de contraintes se définit communément comme étant un ensemble de variables $x_1, \ldots, x_n$, chacune ayant une valeur incluse dans un domaine fini $D_1, \ldots, D_n$, ainsi qu'un ensemble de k contraintes $C_1, \ldots, C_k$ attribuées à chacune des variables $x_1, \ldots, x_n$. La résolution d'un tel

problème consiste à trouver une solution permettant d'attribuer une valeur pour chaque variable en respectant toutes les contraintes imposées. Dans ses travaux, Jung propose d'utiliser ce modèle de variables et de contraintes afin de représenter les propositions et les arguments des agents lors de la négociation; chaque proposition sera représentée par une variable et chaque argument sera représenté par une contrainte. Grâce à cette méthode, le problème de négociation par argumentation peut donc se réduire à un problème bien connu de satisfaction de contraintes distribuées. L'avantage principal de cette approche est de réduire considérablement le temps nécessaire pour converger vers un compromis lors de l'argumentation. De plus, l'algorithme de satisfaction de contraintes utilisé est un algorithme d'optimisation heuristique, il est donc possible de régler la qualité du compromis en fonction du temps que l'on désire allouer pour le calcul de celui-ci. La méthode de Jung s'avèrera donc particulièrement efficace pour les systèmes de coopération comprenant beaucoup d'agents.

D'autres approches ont également été suggérées afin de trouver un modèle formel pour la négociation par argumentation. Ces modèles étaient pour la plupart basés sur un formalisme logique déjà existant. Par exemple, Kraus proposa un modèle s'appuyant sur la logique modale [Kraus et al., 1998] et Sawamura proposa un modèle basé sur la logique dialectique [Sawamura et al., 2000]. Ces deux approches consistaient à modéliser de façon formelle les propositions et les arguments des agents. Cependant, les méthodes utilisant un formalisme logique se sont avérées très lourdes en calcul et difficiles à utiliser dans de gros systèmes multi-agents.

Le modèle de négociation par argumentation a été utilisé pour concevoir plusieurs applications concrètes. Cette technique fut notamment utilisée pour le développement d'un système de senseurs distribués [Jennings et al., 1998].

La négociation par argumentation peut être facilement adaptée dans une multitude de domaines d'application, ce qui constitue un avantage majeur. D'autre part, cette méthode permet la résolution de conflits complexes sans l'intervention d'un arbitre ou d'une instance supérieure du système, ce qui confère plus d'autonomie aux agents et ce qui permet aussi d'éviter les engorgements d'un traitement centralisé. Cependant, cette approche s'avère moins bénéfique lorsque la coopération s'effectue entre un grand nombre d'agents, car dans cette situation, les agents doivent évaluer et concevoir un bon nombre de propositions et de contre-propositions avant d'arriver à un compromis [Jung et al., 2001]. Par ailleurs, les modèles de négociation par argumentation basés sur un formalisme logique sont à l'heure actuelle trop lourds et ne fournissent pas un modèle adéquat pour la résolution de conflits dans un système multi-agents à grande échelle. Finalement, tous les modèles de négociation supposent que les agents présents dans le système utilisent le même langage et les mêmes termes pour décrire une même action. Il serait donc difficile d'implanter un tel modèle dans un système multi-agents hétérogène.

2.7.4 La reconnaissance de plans

Les recherches récentes en intelligence humaine ont démontré que dans la plupart des situations d'interactions, l'être humain utilise les informations qu'il connaît à propos

des autres afin de déduire leurs intentions [Carberry, 2001]. Par exemple, si une personne vous demande où est située la plus proche succursale du service postal Federal Express, et que cette personne vous demande également quels sont les tarifs d'envoi à l'extérieur du pays, vous déduirez que cette personne souhaite probablement envoyer de façon rapide un colis ou une lettre dans un pays étranger. Cette capacité à inférer les intentions des autres nous permet de pouvoir plus facilement les aider, facilite la compréhension mutuelle et par le fait même, favorise la coopération.

Connaissant ces faits, plusieurs scientifiques ont tenté de mettre au point des méthodes permettant de doter les agents de cette faculté de déduction. L'idée de base est de donner à un agent la capacité de reconnaître les intentions d'un autre, qui constituent son plan d'actions, simplement en observant son comportement. Cette technique est appelée la *reconnaissance de plans* et fait l'objet de recherches à l'heure actuelle.

Le modèle de base de cette technique propose qu'on incorpore une base de connaissances à un agent. Cette base sert à contenir un ensemble d'actions $a_1, \ldots, a_n$ et un ensemble de buts $g_1, \ldots, g_n$ en relation avec ces actions [Carberry, 2001]. Ces connaissances sont organisées sous forme de graphe, tel qu'on peut le voir sur la figure 2.5, où chaque action permet d'atteindre un certain but. Ensuite, lorsqu'un agent observe les actions d'un autre, il peut déduire les objectifs de celui-ci en se référant à sa base de connaissances pour interpréter les actions. Par exemple, si un agent en observe un autre poser les actions

subséquentes a_1, a_4 et a_5, il pourrait inférer à l'aide de sa base de connaissances que celui-ci veut sûrement atteindre l'objectif g_6.

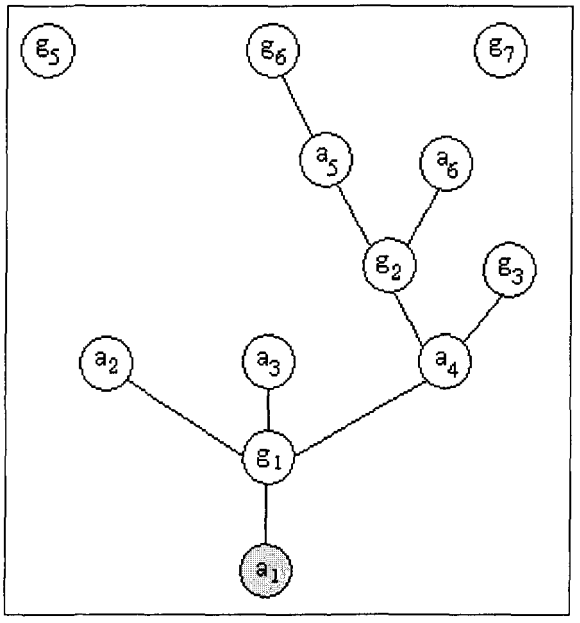

Fig. 2.5 : Représentation de la connaissance pour l'inférence de plans

Les premiers systèmes utilisant ce modèle ont vu le jour à la fin des années 70 et au début des années 80. On pense notamment à deux systèmes parmi les plus connus, soit le système de compréhension d'histoires conçu par Wilensky [Wilensky, 1978] et le système d'interprétation d'expressions de Allen [Allen, 1980]. Ces systèmes étaient cependant très limités. Par exemple, le système de Wilensky ne se préoccupait que du chemin d'inférence minimal dans le graphe en négligeant les autres et le système de Allen, quant à lui, ne fonctionnait qu'avec un nombre d'objectifs très limité.

Le modèle de base fonctionnait bien avec un nombre faible d'actions et de buts, cependant lorsque ce nombre augmentait, la quantité d'hypothèses concernant le nombre d'intentions possibles pour un agent augmentait aussi, causant beaucoup de problèmes de complexité lors des évaluations. Par ailleurs, même si un grand nombre d'hypothèses sont possibles, seulement un nombre restreint sont probables. De plus, il est évident que certaines hypothèses sont plus probables que d'autres. L'amélioration du modèle de base pour la reconnaissance de plans passait donc inévitablement par le développement de méthodes permettant d'éliminer les hypothèses improbables et d'identifier celles qui sont les plus intéressantes [Carberry, 2001].

Durant les années 90, plusieurs solutions ont été avancées afin de résoudre cette problématique. L'une des approches envisagée consistait à demander des précisions sur certaines informations afin de mieux cibler les bonnes hypothèses [Wu, 1991]. Cela implique qu'un agent ne se contenterait plus seulement des informations dont il dispose sur un autre agent, il pourrait également l'interroger afin d'obtenir certaines précisions. Cette idée fut tout de suite reprise et implantée par Cohen et Beek [Cohen et al., 1991]. Leur implantation de cette méthode permettait de restreindre le nombre d'hypothèses plausibles en offrant la possibilité à l'agent d'interroger l'utilisateur. Cependant, il est parfois coûteux en temps de demander des précisions et dans certaines situations, cela peut être carrément impossible. D'autres idées ont donc émergé, la plupart se basant sur des approches probabilistes et sur des méthodes heuristiques. C'est notamment le cas de Raskutti et Zukerman [Raskutti et Zukerman, 1991] qui ont proposé une approche basée sur la

méthode heuristique de la recherche par voisinage. Leur technique utilisait un système de probabilités distribuées à travers l'ensemble des hypothèses afin de converger vers la plus probable.

Récemment, les derniers développements concernant la reconnaissance de plans ont été basés sur l'idée qu'un tel système deviendrait beaucoup plus performant s'il était adapté en fonction de l'agent à observer. Lesh [Lesh, 1997] a d'ailleurs démontré cette augmentation de performance dans ses travaux. Une des méthodes proposées pour cette adaptation est la création de stéréotypes comportementaux basés sur les actions passées de l'agent observé [Ardisonno et Sestero, 1996]. L'idée derrière cette approche est d'observer pendant un certain temps les actions qu'effectue un agent afin de déterminer les séquences répétitives. Par la suite, l'agent qui observe devra inférer ses hypothèses en interprétant la ressemblance entre les actions effectuées et les séquences connues. Cette technique fut notamment utilisée par Bauer [Bauer, 1996] pour reconnaître les préférences d'un utilisateur dans un système de courrier électronique, ainsi que par Carberry et Lambert [Carberry et Lambert, 1999] pour la reconnaissance d'actions de communication.

Bien que l'intérêt de la méthode de reconnaissance de plans en intelligence artificielle ait été démontrée dans de nombreuses applications, elle comporte toutefois plusieurs problèmes qui l'empêchent d'être utilisée dans des applications réelles de grandes envergures [Carberry, 2001]. Le problème le plus sérieux est la robustesse de cette technique par rapport aux erreurs lors de l'entrée d'informations. Si la perception de l'agent

est altérée, le système de reconnaissance de plans va automatiquement être affecté de manière significative. D'autre part, la difficulté à déterminer en toute objectivité laquelle des hypothèses est la plus probable constitue également un problème épineux. Finalement, il peut s'avérer très complexe d'implanter la reconnaissance de plans dans un système multi-agents où les domaines d'expertise de ceux-ci sont hétérogènes. Cela nécessiterait de conserver une base de connaissances contenant un grand éventail d'informations à propos des actions et des objectifs concernant les agents qui oeuvrent dans les différents domaines.

2.7.4.1 La reconnaissance de plans basée sur la logique terminologique

Dans les dernières années, certains chercheurs ont proposé l'utilisation de la logique terminologique pour la représentation et la reconnaissance de plans dans les systèmes multi-agents [Devanbu et Litman, 1991] [Weida et Litman, 1992] [Weida, 1993]. L'idée de base consiste à créer la base de connaissances servant pour la reconnaissance de plans à l'aide d'un système de représentation de la connaissance terminologique. Par exemple, on pourrait définir à l'aide d'une série de concepts l'ensemble des actions que peut effectuer un agent, comme nous le montre la figure 2.6.

Dans cet exemple, la lettre « C », qui précède chacune des étiquettes, signifie que celle-ci représente un concept. La base de connaissances terminologiques de la figure 2.6 pourrait ensuite servir à la conception d'un plan d'actions, tel que nous le montre la figure 2.7.

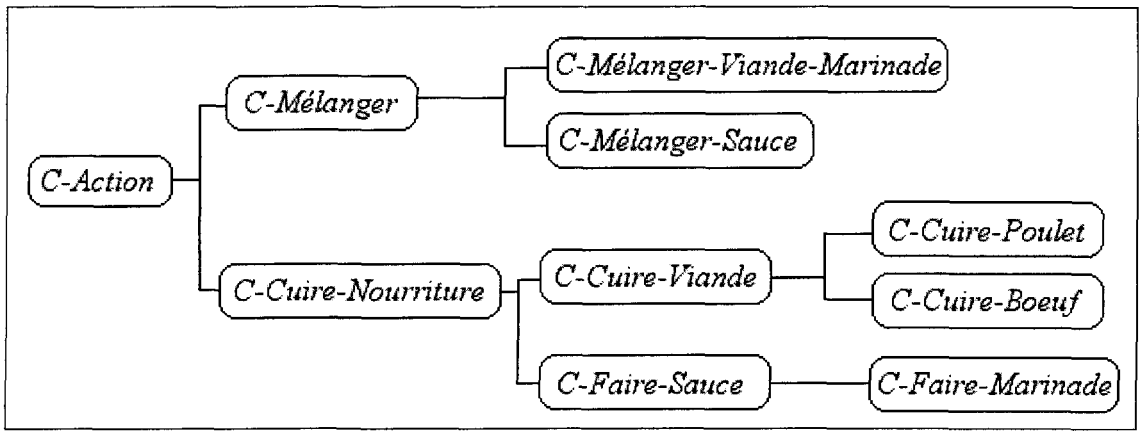

Fig. 2.6 : Taxonomie des actions possibles d'un agent, d'après Weida [Weida, 1993]

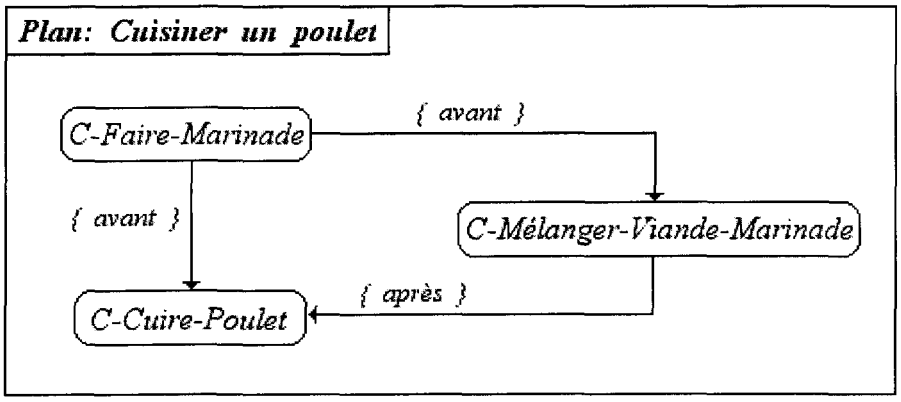

Fig. 2.7 : Plan d'actions terminologique simple, d'après Weida [Weida, 1993]

Les relations entre les concepts de la terminologie représentent des opérateurs temporels simples. Le plan d'actions, dans son ensemble, signifie que l'agent désirant faire cuire un poulet devra confectionner la marinade et en enduire le poulet avant de le faire cuire. Ce plan pourrait être traduit en une série de définitions en logique terminologique, tel que nous le montre la figure 2.8.

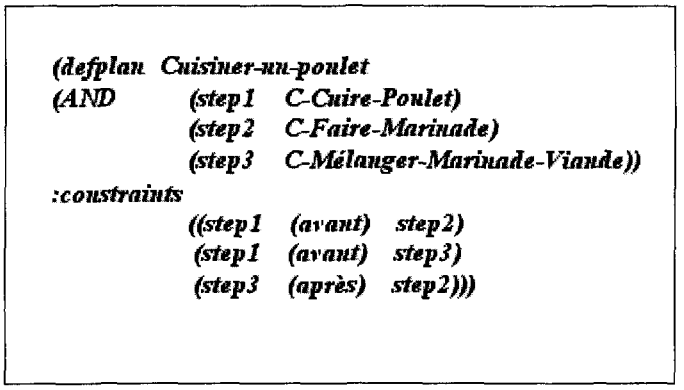

Fig. 2.8 : Un plan d'actions en logique terminologique, d'après Weida [Weida, 1993]

Lorsqu'un agent *A* tente d'effectuer la reconnaissance du plan d'actions d'un agent *B*, il lui suffit de se référer à sa base de connaissances terminologiques afin d'identifier les actions qui ont été posées par celui-ci. Ensuite, l'agent *A* pourra créer, grâce à ses observations, un plan d'actions partielles représentant les actions déjà effectuées par l'agent *B*. Ce plan d'actions sera construit à l'aide d'une définition terminologique d'un nouveau concept de plan. Ce nouveau concept, représentant le plan d'action partiel de l'agent, sera automatiquement classifié par le Système de Représentation des Connaissances Terminologiques (SRCT), comme nous le verrons au chapitre 3, dans la taxonomie de concepts déjà existants. L'agent *A* devra préalablement posséder une série de scénarios de plans possibles, tel que celui de la figure 2.7. En définitive, il suffira à l'agent *A* de questionner sa base de connaissances terminologiques pour savoir à quel endroit le nouveau concept a été positionné, comme nous le montre la figure 2.9. La position de celui-ci permettra d'inférer facilement quelles sont les possibilités d'intentions de l'agent *B* en fonction des actions qu'il a déjà effectuées.

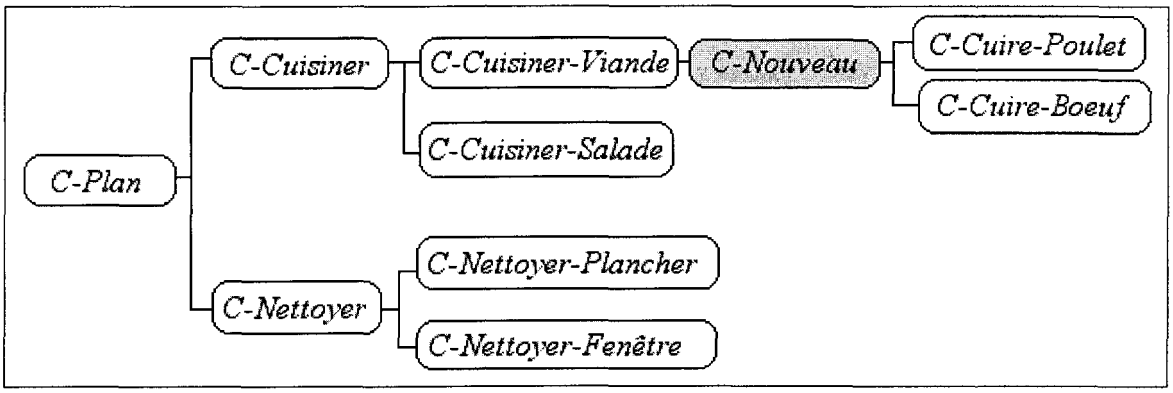

Fig. 2.9 : Classification d'un plan partiel à travers une taxonomie de plan

L'utilisation d'une base de connaissances terminologiques pour la représentation et la reconnaissance de plans comporte un avantage majeur: elle permet d'inférer de façon efficace les possibilités d'actions d'un agent, grâce à l'algorithme de classification par subsomption. Cette particularité évite d'avoir à parcourir un arbre gigantesque de possibilités. Elle permet également de reconnaître deux actions identiques ne portant pas le même nom, grâce à leurs définitions terminologiques. De plus, la création de l'arborescence des actions s'effectue de façon automatique lorsqu'on introduit les définitions de celles-ci. Cependant, la performance de la classification, donc de la reconnaissance de plans, est directement liée au SRCT utilisé. L'expressivité du langage terminologique utilisé limitera également les possibilités de définitions des actions.

2.7.5 La délégation

La délégation s'applique principalement aux tâches. Un agent peut vouloir déléguer une ou plusieurs tâches à un autre agent, dans un geste de coopération [Haddadi, 1996]. Par

exemple, un agent qui aurait une tâche de calcul mathématique complexe à effectuer pourrait faire appel à un autre agent, spécialisé dans ce type de calcul. Cette forme de délégation est de loin la plus répandue. Il est également possible de parler de délégation de rôles [Werner, 1990]. Un agent peut avoir un statut hiérarchique dans un regroupement d'agents, ce qui constitue son rôle social. Ce rôle peut donc aussi être délégué, de façon temporaire ou permanente, à un autre agent. Finalement, il est possible d'effectuer une délégation de contrôle. Lorsqu'un agent est en contrôle du déroulement d'une situation, il peut déléguer ce contrôle à un autre agent pour diverses raisons, sans pour autant lui céder son rôle.

Un bon nombre de travaux dans ce champ de recherche ont été effectués par Castelfranchi et Falcone [Castelfranchi et Falcone, 1997], [Castelfranchi et Falcone, 1998], [Castelfranchi et Falcone, 2000], qui sont à l'origine du modèle de la *délégation* et de l'*adoption* de buts basé sur les plans [Castelfranchi et Falcone, 1998]. Ce modèle suggère que lors d'une délégation de tâches, celle-ci soit prise en compte dans l'état mental de l'agent délégant. Étant donné qu'une action sert toujours à atteindre un certain état, qui peut être perçu comme un but, la délégation d'une action implique nécessairement l'adoption par un autre agent d'une paire action/but noté $\tau = (a, g)$. Par le symbole τ, on fait ici référence à une action a, qui produit comme résultat un but g. En définitive, cela signifie qu'un agent A_1 qui délègue une action ou qui a besoin du résultat d'une action exécutée par un agent A_2, inclura cette action dans son propre plan d'actions comme s'il l'exécutait lui-même, tel qu'on peut le voir sur la figure 2.10. L'agent A_1 délègue une partie de son plan

d'actions, c'est-à-dire la réalisation des actions a_3 et a_5 à l'agent A_2. Cependant, on note que l'agent A_1 garde l'exécution de celles-ci à l'intérieur de son propre plan d'actions. On peut également constater que l'agent A_2 adopte une certaine partie de l'objectif g de l'agent A_1, c'est-à-dire l'atteinte de l'état σ_6. On pourra donc dire que l'agent A_2 a adopté une certaine partie du plan d'actions de A_1.

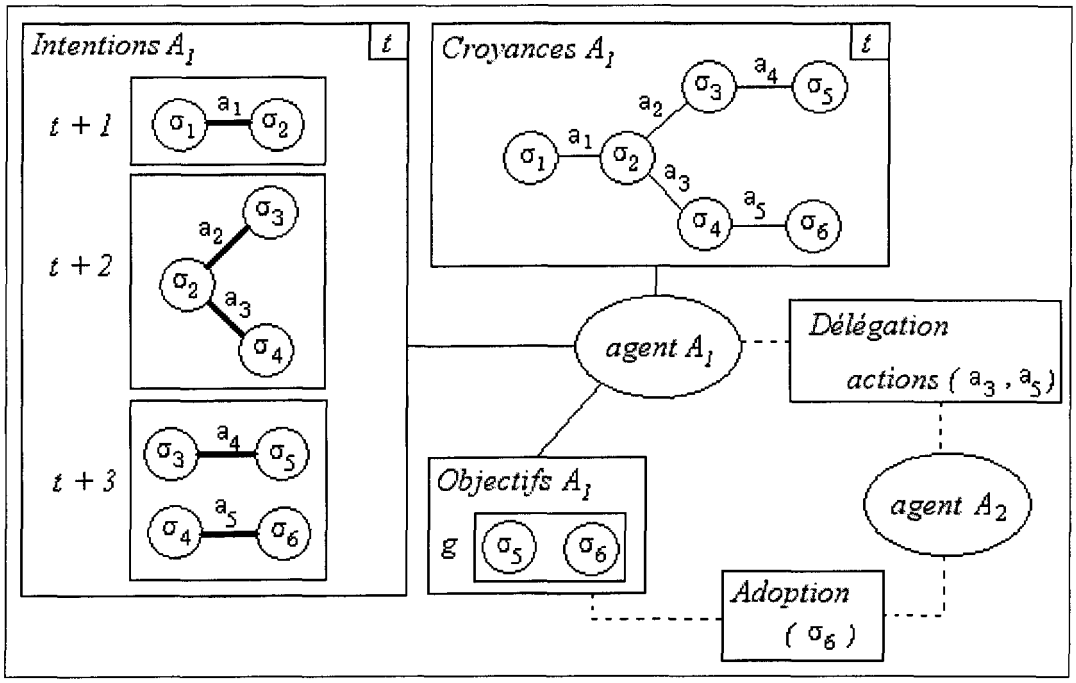

Fig. 2.10 : Modèle de délégation et d'adoption

Le modèle de Castelfranchi et Falcone procure beaucoup d'avantages, notamment celui de ne pas nécessiter l'intervention d'un arbitre ou d'un organisateur pour la coopération par délégation de tâches. Il permet également l'établissement d'une certaine base de connaissances partagée entre les agents coopérants. Cette approche favorise l'autonomie des agents et décentralise la gestion de la délégation des tâches. Il est

également à noter que cette technique est l'une des seules qui tienne compte de l'état mental des agents lors de la délégation de tâches. L'incorporation des objectifs délégués dans l'état mental des agents permet à ceux-ci de pouvoir effectuer des prévisions en fonction des objectifs communs.

Malgré plusieurs avantages, il ne faut pas oublier que les agents faisant partie d'un tel système doivent avoir la volonté de coopérer et d'accepter d'effectuer des tâches afin d'aider les autres agents. Pour cela, ils doivent avoir les mêmes objectifs ou avoir certains objectifs similaires. D'autre part, même quand deux agents ont les mêmes buts, il se peut qu'ils aient des points de vue (opinions) différents sur la manière de les atteindre [Tessier et al, 2001]. Cela entraînera inévitablement des conflits lors de la délégation de tâches, causés par ces divergences d'opinions. Par exemple, si un agent A_1 veut confier une tâche qu'il considère simple à un agent A_2, et que celui-ci pense que cette tâche est complexe, il en résultera un conflit, comme nous le montre la figure 2.11. Le symbole S_{Ax} représente l'ensemble des compétences d'un agent A_x, $B\ Act_{Ax}$ signifie l'ensemble des actions élémentaires pour un agent A_x, $C\ Act_{Ax}$ représente l'ensemble des actions complexes pour un agent A_x et finalement, le symbole $NR\ Act_{Ax}$ signifie l'ensemble des actions pour lesquelles un agent A_x ne possède pas de règles de réduction. On peut donc interpréter la première case en haut à gauche du tableau comme suit : si un agent A_1 a les compétences nécessaires pour exécuter une action a ou bien qu'il évalue cette action comme étant de base, alors il croira que cette action est élémentaire. De son côté, si l'agent A_2 croit la même chose, il n'aura aucun conflit, cependant s'il évalue que l'action a est complexe ou bien s'il

ne possède aucune règle lui permettant de réduire cette action à une action de base, il croira que cette action est complexe et sera en conflit avec l'agent A_1.

Point de vue A_1	***Point de vue correspondant*** A_2	
$\tau = a$ *avec* $(a \in S_{A_1}) \vee (a \in BAct_{A_1})$ *"C'est une action élémentaire"*	$\tau = a$ *avec* $(a \in S_{A_2}) \vee (a \in BAct_{A_2})$ *"C'est une action élémentaire"* — *Pas de conflit*	$\tau = a$ *avec* $(a \in CAct_{A_2}) \vee (a \in NRAct_{A_2})$ *"C'est une action complexe"* — *Conflit*
$\tau = a$ *avec* $(a \in CAct_{A_1}) \vee (a \in NRAct_{A_1})$ *"C'est une action complexe"*	$\tau = a$ *avec* $(a \in S_{A_2}) \vee (a \in BAct_{A_2})$ *"C'est une action élémentaire"* — *Conflit*	$\tau = a$ *avec* $(a \in CAct_{A_2}) \vee (a \in NRAct_{A_2})$ *"C'est une action complexe"* — *Pas de conflit*

Fig. 2.11 : Conflits causés par la divergence des points de vue entre deux agents

Cet aspect conflictuel constitue un problème réel pour l'application à grande échelle du modèle. Afin de gérer un tel modèle de coopération, il est nécessaire pour les agents de pouvoir comprendre les objectifs, les plans et les intérêts des autres agents du système [Tessier et al., 2001]. Autrement dit, cette méthode ne peut s'appliquer qu'entre des agents utilisant la même sémantique pour décrire leurs points de vue. Malgré ces problèmes, il est évident qu'un niveau élevé de coopération dans un système multi-agents requiert l'implantation d'un modèle quelconque de délégation [Castelfranchi et Falcone, 1997]. Le modèle de Castelfranchi et Falcone se veut donc un cadre de base formel pour le

développement de méthodes de délégation plus complètes qui incluront des mécanismes pour la résolution des conflits.

2.7.6 Planification à initiative mixte : l'usager dans la boucle

L'intégration d'un usager dans un système mutli-agents a toujours constitué un défi de taille et ce, depuis l'apparition de ce type de système. La grande question entourant cette problématique est : comment intégrer l'utilisateur dans la même boucle de réalisation (planification, exécution) d'une tâche? En d'autres termes, la réalisation d'une tâche conjointe usager-agent ne devra pas être uniquement l'œuvre de l'une ou l'autre des entités, mais plutôt de l'ensemble des deux - usager dans la boucle -, dans lequel ils auront les mêmes possibilités de proposer, de suspendre, de refuser et d'arrêter l'autre [Dautenhahn, 2001]. Récemment, les recherches sur la cohabitation usager-agent ont pu apporter quelques réponses partielles à cette problématique [Cesta et D'Aloisi, 1999] [Castelfranchi et Falcone, 2000] [Bouzouane et Bouchard, 2003].

L'autonomie de l'agent est la raison principale qui empêche l'utilisateur d'accepter facilement la cohabitation avec celui-ci [Norman, 1994]. En fait, l'usager craint de perdre le contrôle lors d'une situation d'interaction au profit de l'agent ou bien de ne pas pouvoir intervenir lorsqu'il le désire. Cet aspect de la problématique d'intégration rend la tâche particulièrement difficile. Ensuite, une autre problématique importante est générée par le fait qu'il est très difficile de modéliser l'usager[Tessier et al., 2001]. D'autre part, il existe un problème de divergence paradoxal entre la notion d'autonomie et de contrôle. Ce

problème est d'ailleurs au cœur de la controverse sur l'intégration usager-agent et résulte du contraste existant entre une manipulation directe de l'agent via une interface et l'autonomie de celui-ci. Par ailleurs, certains systèmes complexes, tel que le pilotage d'un avion, ne peuvent pas fonctionner continuellement en mode manuel ou automatique. Les opérations manuelles impliquent des risques d'erreurs, car l'utilisateur n'est pas infaillible. D'un autre côté, les opérations automatiques, qui sont sous la responsabilité d'un agent autonome, sont très risquées, car un disfonctionnement de l'agent peut rendre le système non opérationnel. Certains chercheurs plaident en faveur d'un transfert de contrôle à l'usager lorsque la situation excède les capacités de l'agent, mais le problème est que l'utilisateur peut être submergé par l'arrivée massive d'informations inhérentes à un système complexe [Kitano, 1996]. En définitive, l'intégration usager-agent doit faire face à plusieurs questions, telles que: quand faut-il passer le contrôle à l'usager ou à l'agent pour contribuer efficacement à la résolution de problèmes et comment savoir si les actions de l'entité qui prendra le contrôle seront efficaces?

L'une des approches les plus souvent préconisées pour l'intégration usager-agent est celle de la *planification par initiative mixte* [Rich et al, 2000]. Une initiative signifie prendre une décision sur ce qu'on doit faire dans le futur [Hearst et al., 1999]. Ceci n'implique pas que cette décision sera effectuée par l'entité qui a pris cette initiative. Par exemple, un usager pourrait exécuter un plan d'actions proposé par l'initiative d'un agent. En d'autres termes, l'initiative est bi-directionnelle. La planification par initiative mixte est donc une technique de planification d'actions consistant à faire travailler un usager et un

agent de façon parallèle, en permettant aux deux d'intervenir à n'importe quel moment durant les différentes étapes de réalisation de la tâche, comme nous le montre la figure 2.12.

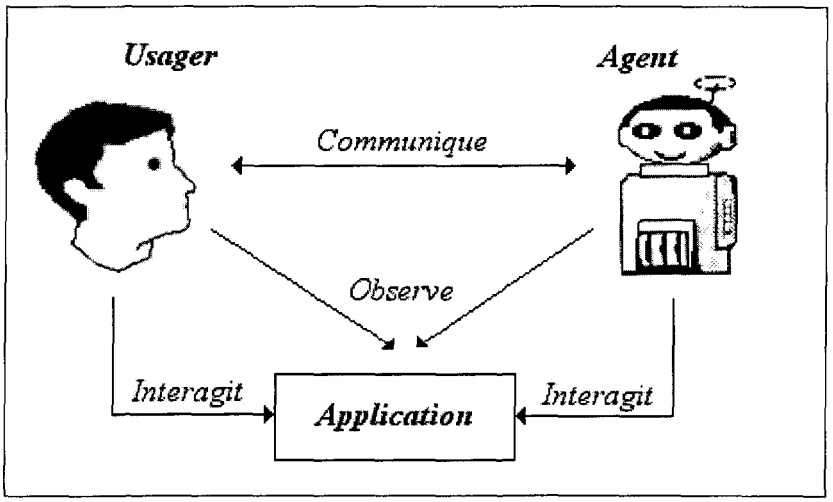

Fig. 2.12 : Modèle de planification à initiative mixte

Dans un système suivant le modèle de planification à initiative mixte, une action ne pourra avoir lieu que lorsque les deux entités seront d'accord sur son exécution [Rich et al, 2000]. La façon de procéder est simple, à tout moment l'une des deux entités est en contrôle des décisions sur les actions futurs (contrôle de l'initiative). De façon parallèle, l'autre entité peut à n'importe quel moment intervenir sur une décision prise par celle en contrôle de l'initiative. L'entité qui n'est pas en contrôle peut également effectuer, au moment souhaité, une proposition de plan d'actions futur, comme nous le montre la figure 2.13. Si cette proposition est acceptée, un transfert de contrôle de l'initiative sera effectué. Lorsque les deux entités ne sont pas d'accord sur une proposition de plan d'actions, ils

peuvent effectuer une négociation afin d'arriver à une entente commune. L'action proposée pourra alors être effectuée. Sinon, la proposition sera tout simplement refusée.

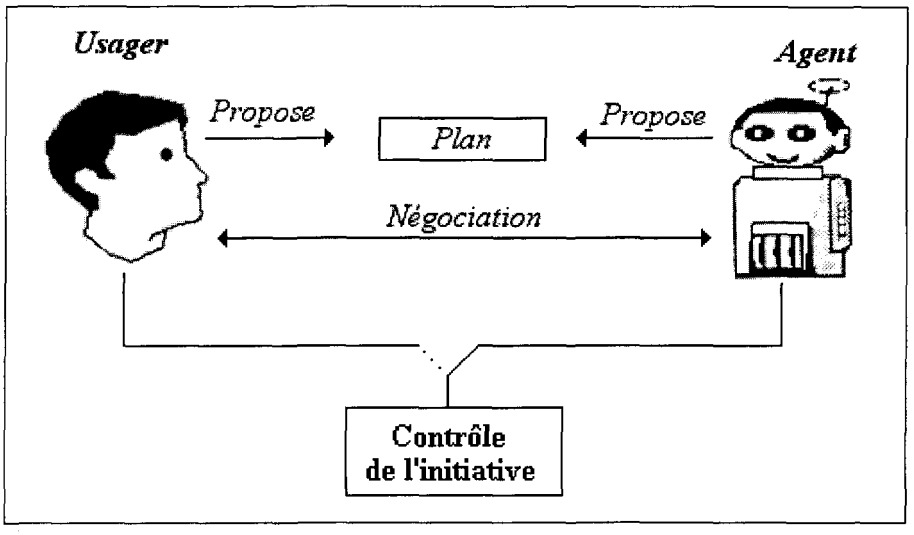

Fig. 2.13 : Exemple d'une planification à initiative mixe

Une approche intéressante pour l'implantation d'un modèle à initiative mixte a été suggérée par Cesta et d'Aloisi [Cesta et D'Aloisi, 1999], lors du développement du système *Masma* (Multi Agents System for Meeting Automation). Ce système a été conçu afin permettre à un utilisateur de rendre plus efficace la gestion de son emploi du temps (agenda) en déployant un agent agissant en son nom pour l'aider à gérer ses rendez-vous d'affaires. Cet agent peut par la suite dialoguer avec d'autres agents représentant d'autres hommes d'affaires et ainsi proposer des ajouts à l'horaire de son usager. L'utilisateur peut lui aussi proposer des ajouts à son horaire. La différence majeure du modèle de Cesta et d'Aloisi réside dans leurs méthodes de délégation du contrôle de l'initiative. Leur stratégie

consiste à déléguer le contrôle de l'initiative usager-agent en fonction de la situation actuelle, des connaissances partagées par les deux entités et du profil de l'utilisateur. La décision du transfert de contrôle s'effectue grâce à une méthode de calcul heuristique qui prend en compte ces différents paramètres.

L'avantage du modèle de Cesta et d'Aloisi est de donner une méthode concrète permettant de déterminer quand il devra y avoir un changement de contrôle de l'initiative. Cependant, leur modèle ne prend pas en compte certaines réalités. Par exemple, il est possible que l'usager interprète mal ou ne comprenne pas les décisions et les actions de l'agent. Cela ne signifie pas pour autant que celles-ci sont mauvaises ou non justifiées. L'agent pourrait simplement avoir un plan d'actions différent, et peut-être même meilleur que celui de l'utilisateur. Cette situation entraînera forcément une intervention de l'usager, qui jugera que l'agent n'agit pas correctement. De plus, le transfert d'initiative est dépendant du domaine d'application.

Le problème de la coopération usager-agent via une planification à initiative mixte est donc beaucoup plus complexe qu'il y paraît, car en plus de considérer l'aspect de la confiance de l'usager envers son agent, il faut également tenir compte de la notion d'équilibre des pouvoirs entre les deux. Cet équilibre permettra d'assurer la validité des initiatives prises par chacune des entités. Pour cela, l'utilisateur et l'agent devront avoir une confiance mutuelle l'un envers l'autre. Afin d'atteindre cet objectif, l'agent et l'usager doivent pouvoir se comprendre et comparer leurs différentes visions ou points de vue sur la

manière d'agir. Par ailleurs, un des obstacles majeurs empêchant le couple usager-agent de bien se comprendre est la diversité terminologique de ceux-ci. Ce problème vient du fait que l'usager et l'agent peuvent utiliser des termes totalement différents pour décrire une même situation. Par exemple, un utilisateur pourrait affirmer qu'il souhaite « liquider toutes ses parts de la compagnie » et l'agent pourrait affirmer qu'il veut « vendre la totalité de ses actions de l'entreprise ». Ces deux affirmations sont constituées de termes différents qui signifient la même chose. L'aspect terminologique est au cœur des problèmes de compréhension lors des interactions dans une société d'agents et particulièrement entre un utilisateur et son agent agissant en son nom [Bouzouane et Bouchard, 2003]. Cette problématique se complique encore davantage lorsque les points de vue des agents sont différents et qu'ils ne sont pas totalement équivalents. Dans cette situation, il est légitime de se demander, en tenant compte de la diversité terminologique, dans quelle mesure ces points de vue sont équivalents ou divergents? Nous appellerons donc les divergences de points de vue, causées par des problèmes de compréhension terminologique, des conflits terminologiques. Ces conflits ne sont pas pris en considération par les autres méthodes de coopération citées dans les sections précédentes.

2.8 Bilan sur les méthodes de coopération

Jusqu'à maintenant, aucune proposition n'a été faite sur la manière de comparer et de mesurer la divergence entre les différents plans d'actions d'un usager et d'un agent. Falcone et Castelfranchi [Castelfranchi et Falcone, 2000] arrivent pourtant à la conclusion qu'il est nécessaire de développer une théorie analytique afin de superviser la délégation du

contrôle lors d'une planification à initiative mixte. De plus, aucune approche tenant compte du problème de diversité terminologique n'a encore été suggérée. En définitive, une question fondamentale demeure toujours sans réponse: de quelle façon peut-on mesurer, comparer et combiner ensemble deux points de vue usager-agent afin d'assurer une bonne planification à initiative mixte [Falcone et Castelfranchi, 2001]?

2.9 Conclusion

Ce chapitre nous a permis de faire l'état de l'art sur la problématique entourant la coopération dans les systèmes multi-agents (SMA). Cela a été réalisé grâce à une présentation des définitions existantes sur les notions d'agents et de systèmes multi-agents, suivie d'une description approfondie entourant les méthodes de coopération dans une société d'agents, pour finalement terminer avec une revue littéraire des récents travaux entourant le domaine. Dans la majorité des cas, les explications ont été suivies d'exemples concrets permettant de mieux comprendre l'ensemble du domaine. De plus, ce chapitre a permis de situer de façon beaucoup plus précise le cadre de nos travaux, qui s'inscrivent dans la lignée des recherches sur la coopération usager-agent.

Dans le cas de notre étude, nous nous intéressons particulièrement aux conflits liés aux divergences entre les points de vue d'un utilisateur et d'un agent, pour garantir une prise d'initiative mixte dans un contexte de coopération. Ainsi, on peut affirmer que deux points de vue sont divergents lorsqu'ils ne sont ni identiques, ni contradictoires. Par exemple, lorsqu'un agent croit qu'il fera beau demain et qu'un autre croit qu'il fera

mauvais, leurs points de vue sont contradictoires, car le fait qu'il fasse mauvais implique qu'il ne fera pas beau. Cependant, si un agent croit qu'il fera beau demain et qu'un autre croit qu'il y aura du soleil avec quelques petits nuages, leurs points de vue ne sont pas contradictoires, ni identiques, ils sont donc divergents. Ce type de situation constitue un problème épineux, car il est très difficile de définir dans quelle mesure deux points de vue peuvent être divergents ou semblables. On se trouve donc dans une zone grise qui peut nous amener à nous poser une multitude de questions : sur quelles notions les points de vue des agents divergent-ils? Est-ce que cette divergence est significative? Est-ce que cette divergence peut nuire à une coopération éventuelle?

D'autre part, les modèles actuels de coopération ne supportent pas la gestion des conflits terminologiques. Pourtant, il a été démontré qu'il était nécessaire d'arriver au développent d'une théorie analytique permettant de tenir compte de cet aspect lors de la résolution des conflits, notamment pour la délégation [Castelfranchi et Falcone, 2000].

Nous proposerons donc au chapitre 4 une approche analytique novatrice permettant d'effectuer une comparaison objective des points de vue d'un utilisateur et d'un agent. Nous proposerons une méthode pour mesurer la similarité entre les points de vue de ceux-ci, basée sur la logique terminologique, qui sera décrite au chapitre suivant. Nous verrons comment l'utilisation de cette logique permettra de prévenir les problèmes liés aux conflits terminologiques. Cette méthode facilitera la compréhension usager-agent et minimisera les

communications. Finalement, nous validerons notre approche avec une implantation concrète de celle-ci dans une application de gestion de stocks en commerce électronique.

CHAPITRE 3

LA LOGIQUE TERMINOLOGIQUE

3.1 Introduction

Le domaine de l'intelligence artificielle est composé d'un grand nombre d'axes de recherche qui ont été engendrés par plusieurs problématiques de natures diverses. L'un des premiers problèmes fut la nécessité de représenter et de manipuler la connaissance. Cette problématique ouvrit la voie à la recherche sur les différentes façons de la formaliser de manière flexible, simple et de façon à pouvoir représenter le plus grand nombre de réalités possibles. La conception d'un système intelligent passe donc obligatoirement par le choix d'un formalisme adéquat de représentation des connaissances qui lui permettra à la fois de décrire celle-ci et de pouvoir en déduire d'autres connaissances. Ce chapitre présentera l'un de ces formalismes de représentation appelé logique terminologique ou logique descriptive. Ce type de logique est particulièrement bien adapté pour gérer l'interprétation et la classification des connaissances. Cela peut se traduire concrètement par la conception d'ontologies ou la mise en place de systèmes de représentations de la connaissance terminologique (SRCT).

Dans un premier temps, nous définirons la notion de connaissance afin de mieux comprendre la complexité et l'importance des problèmes liés à sa représentation. Ensuite, la deuxième partie portera sur la théorie concernant la logique terminologique, incluant les concepts de base sur lesquels repose ce formalisme ainsi que les différentes notations de celui-ci. Finalement, la dernière section du chapitre portera sur l'architecture d'un SRCT et fera une présentation de l'un des plus connus.

3.2 La notion de connaissance

Afin de mieux situer les origines de la logique terminologique, il faut d'abord définir la notion de connaissance, telle qu'elle s'est développée en intelligence artificielle (IA). Elle désigne l'ensemble des informations (savoir, savoir-faire, expériences, souvenirs, concepts, faits...) nécessaires à un être humain (ou à une machine) de manière à ce qu'il puisse accomplir une tâche considérée comme complexe. Par exemple, diagnostiquer une maladie, résoudre un problème de mathématique, réparer une machine, traiter un cas juridique, gérer un ensemble de comptes financiers, faire une campagne de marketing ou traduire un texte sont des activités, qui, lorsqu'elles sont réalisées par des êtres humains, réclament une grande quantité de compétences et d'informations organisées dans un but précis [Ferber, 1995]. D'un point de vue IA, représenter des connaissances consiste à trouver des structures de données appropriées au stockage et à la manipulation des informations relatives à une application [Hall, 1962] [McCarthy et Hayes, 1969] [Pylyshyn, 1989].

Les recherches effectuées sur les méthodes de représentation efficaces d'un ensemble de connaissances furent divisées en deux axes de recherches bien distincts : d'une part, il y a eu la conception de méthodes graphiques de représentation tels les réseaux sémantiques [Schmidt, 1991] et les langages à base de cadres (frames) [Minsky, 1974] et, d'autre part, les recherches ont surtout été dirigées vers les formalismes basés sur la logique mathématique, telle que la logique du premier ordre [Hamburger et Richards, 2002].

3.3 Théorie concernant la logique terminologique

On peut définir la logique terminologique comme étant un formalisme de représentation des connaissances se divisant en deux parties bien distinctes : la partie terminologique, qui permet de définir un ensemble de concepts ainsi que leurs rôles, et la partie assertionnelle, qui permet la création d'individus appartenant aux différents concepts de la partie terminologique. La partie assertionnelle du formalisme permet les opérations de classification et d'inférence. La classification, qui constitue la base du raisonnement en logique terminologique, s'appuie sur la relation de subsomption qui permet d'organiser les concepts et les rôles en une hiérarchie de termes.

3.3.1 Les éléments de base de la logique terminologique

Une logique terminologique peut être définie comme la conjonction d'un langage terminologique et d'un langage assertionnel. Un langage terminologique peut à son tour être défini comme étant un ensemble d'opérateurs (appelés des constructeurs) et de symboles qui permettent la définition de concepts et de rôles. Par exemple, le langage terminologique *TF*, que nous utiliserons pour les démonstrations tout au long du chapitre, peut être défini comme suit [Nebel, 1995]:

$$TF = < \top, \bot, \doteq, \leq, \textit{disjoint, and, all, atleat, atmost} >$$

où $\{\top, \bot, \doteq, \leq\}$ sont des symboles et $\{\textit{disjoint, and, all, atleat, atmost}\}$ sont des opérateurs. Le langage *TF* sera décrit de façon plus explicite à la section 3.3.2.

Un langage assertionnel peut être défini comme étant un ensemble de symboles et d'opérateurs permettant la création d'instances issues de concepts et de rôles décrits par un langage terminologique. Par exemple, le langage terminologique *AF*, qui est le complément assertionnel de *TF*, se définie comme suit :

$$AF = < c, r, \geq, \leq, atleast, atmost>$$

où $\{\geq, \leq\}$ sont des symboles et *{c, r, atleat, atmost}* sont des opérateurs. Il est à noter que *c* et *r* représentent ici les noms de deux termes désignant respectivement un concept *c* et un rôle *r* décrit dans la terminologie. Les opérateurs et les symboles du langage *AF* seront décrit de façon plus explicite à la section 3.4.2.

3.3.1.1 La notion de concept

D'un point de vue ensembliste, on peut voir un concept comme étant une série de définitions qui permettent la création d'un ensemble en spécifiant les conditions nécessaires d'appartenance à celui-ci. Un concept[1] en logique terminologique peut être vu comme une conjonction de caractéristiques qui sont définies par ses rôles. En fait, il existe deux types de concepts en logique terminologique:

- **Concept primitif** : les concepts primitifs servent de base à la construction de concepts définis. Ce type de concept représente le niveau le plus atomique de la connaissance en logique terminologique. Ces concepts peuvent posséder certains

rôles qui les définissent, mais leurs descriptions sont incomplètes et représentent des conditions nécessaires mais insuffisantes pour déterminer le type de l'individu: il n'est pas possible de reconnaître un représentant d'un concept primitif à la vue de ses seuls rôles [Haton et al., 1991]. Ainsi, si un concept nommé « Personne » et un autre nommé « Animal » sont tous deux définis avec le rôle « date-de-naissance », il sera vrai de dire que toutes les instances du concept « Personne » auront une « date-de-naissance », mais l'inverse ne sera pas vrai.

- **Concept défini** : les concepts définis sont construits à partir des rôles qu'ils entretiennent avec des concepts primitifs ou avec d'autres concepts définis. Ce type de concept représente les niveaux de connaissances les plus complexes. Contrairement aux concepts primitifs, les concepts définis possèdent une définition complète et suffisante pour en déduire l'appartenance certaine d'une instance [Haton et al., 1991]. Si l'on prend, par exemple, un concept nommé « Parent » qui entretient un rôle nommé « enfant », il sera possible d'affirmer qu'une instance de type « Parent » aura automatiquement un ou des « enfant » et qu'une instance ayant un ou des « enfant », appartiendra automatiquement au concept « Parent ».

---

[1] Habituellement, chaque mot désignant le nom d'un concept commence par une lettre majuscule

3.3.1.2 La notion de rôle

Un rôle[2] représente une relation binaire entre deux concepts. C'est une relation qui a pour fonction de caractériser le premier concept en précisant sa relation avec le second. Le concept non caractérisé par le rôle est appelé le co-domaine. Des restrictions de cardinalité peuvent être attribuées aux rôles. Par exemple, le rôle « enfant », caractérisant le concept « Parent », devrait être défini avec une cardinalité $n \geq 1$.

3.3.1.3 Relation entre les rôles

Une relation entre deux rôles est un lien servant à préciser un rôle à l'aide d'un autre. Par exemple, une relation pourrait être introduite entre un rôle « date-de-naissance » et un rôle « date-de-naissance-humain » afin de préciser que le deuxième est une spécialisation du premier. On peut également vouloir préciser que deux rôles sont complètement différents, même s'ils définissent le même concept et ont le même co-domaine. Ainsi, un rôle nommé « père-de » pourrait entretenir un lien avec un rôle « mère-de » afin de le différencier de celui-ci. Il est à noter qu'on ne peut pas attribuer de restrictions sur la cardinalité de ces relations.

3.3.2 Création de concepts et de rôles

La logique terminologique regroupe une grande quantité de langages terminologiques qui ont tous été construits à partir des éléments de base vus précédemment. Afin de mieux comprendre le fonctionnement des langages de descriptions, cette section du

[2] Habituellement, le nom d'un rôle est écrit entièrement en lettres minuscules

chapitre présentera la théorie sur la construction de concepts et de rôles en utilisant le formalisme du langage terminologique TF défini par Nebel [Nebel, 1995]. Ce langage est très proche de celui développé par Brachman et Schmolze [Brachman et Schmolze, 1985] pour KL-ONE. Il est particulièrement simple et facile à comprendre.

3.3.2.1 Construction de concepts avec le langage TF

En logique terminologique, un *terme* désigne le nom d'un concept ou d'un rôle. La terminologie du langage TF est principalement basée sur l'introduction de nouveaux termes et sur l'opération de disjonction. Avant d'aller plus loin dans les explications, il faut d'abord introduire quelques symboles qui serviront tout au long de la présentation.

C_i	=	*Symbole qui représente le terme d'un concept*
R_i	=	*Symbole qui représente le terme d'un rôle*
$\top$	=	*Symbole représentant le concept et le rôle maximal, le « tout »*
$\bot$	=	*Symbole représentant le concept et le rôle minimal, le « rien »*
$\leq$	=	*Symbole servant à définir un concept ou un rôle primitif*
$\doteq$	=	*Symbole servant à définir un concept ou un rôle complexe*

En langage TF, un concept peut être défini de trois façons différentes en fonction de sa nature. Premièrement, il est possible de définir un concept primitif qui ne possède aucune caractéristique et qui servira de base à la création de concepts plus complexes. Ce type de concept doit être défini comme un descendant du concept maximal $\top$. Un concept

primitif peut être défini selon sa position par rapport aux autres concepts de la terminologie en spécifiant la combinaison de concepts qui le compose. Finalement, on peut définir un concept complexe à l'aide d'opérateurs qui seront appliqués sur des concepts déjà existants afin d'en définir ses caractéristiques. Voici les trois différentes notations pour la définition de concepts en TF :

<Concept primitif> $\leq$ T

<Concept primitif> $\leq$ *<concept ou combinaison de concepts>*

<Concept complexe> $\doteq$ *<concept ou combinaison de concepts>*

3.3.2.2 Opérations sur les concepts

Les opérateurs servent spécifiquement à définir les concepts, les combinaisons de concepts et les rôles dans la terminologie. Plus les opérateurs d'un langage permettront de définir de façon précise les concepts et les rôles, plus ce langage pourra représenter un grand nombre de réalités. Il est possible de créer des concepts complexes à l'aide de certains opérateurs (qui sont également appelés des constructeurs). Il est à noter qu'il y a des différences marquées en ce qui concerne les opérateurs des différents langages terminologiques, notamment au niveau du nombre disponible. Par exemple, le langage TF ne possède pas d'opérateur de négation ni d'opérateur existentiel, ce qui limite la flexibilité du langage. La notation habituelle qui est utilisée pour la définition de ces opérateurs est très proche du langage Lisp (Common Lisp). En fait, il est possible de décrire les opérateurs sous forme concrète (Lisp) et sous forme abstraite (logique classique). Voici la liste des opérateurs disponibles en langage TF :

a) **Disjoint** : cet opérateur spécial sert à introduire des disjonctions entre deux concepts. Il peut également être utilisé avec les rôles. Il a pour fonction de signaler que certains concepts ou rôles sont opposés, par exemple les concepts « Homme » et « Femme ». L'utilisation de cet opérateur de disjonction se fait de la façon suivante :

Forme concrète : *(disjoint <Concept C_1> <Concept C_2>)*

Forme abstraite : $C_1 \sqcup C_2$

Exemple TF : *(disjoint Homme Femme)*

b) **AND** : cet opérateur représente le « ET » logique. Il est utilisé pour créer une conjonction de plusieurs concepts. Par exemple, la création d'un concept « Dauphin » pourrait nécessiter la conjonction entre le concept « Animal-Marin » et « Mammifère ». La notation utilisée pour cet opérateur est la suivante :

Forme concrète : *(AND C_1 ... C_n)*

Forme abstraite : $C_1 \sqcap \ldots \sqcap Cn$

Exemple TF : *Dauphin ≐ (AND Animal-Marin Mammifère)*

c) **ALL** : Cet opérateur exprime la restriction de valeurs. En langage TF, il a pour fonction de restreindre la nature du co-domaine d'un rôle donné. Prenons, par exemple, un rôle R_1 dénoté $R_1(C_1, C_2)$. Il serait possible, grâce à l'opérateur *ALL*, de définir la nature exacte du concept C_2 de la relation. En d'autres termes, on pourrait

utiliser cet opérateur dans la définition d'un concept « Dauphin » pour limiter le co-domaine d'un rôle nommé « mange » aux instances de concepts « Nourriture ». La notation de cet opérateur est la suivante :

Forme concrète : *(ALL R C)*

Forme abstraite : $\forall R : C$

Exemple TF : *Dauphin ≐ (AND Animal-Marin Mammifère ALL mange Nourriture)*

d) **ATLEAST** : Cet opérateur permet d'effectuer une restriction de cardinalité minimum sur un rôle donné. Il sert à définir une borne inférieure sur un rôle. Par exemple, un rôle « enfant » d'un concept « Parent » pourrait se voir imposer une cardinalité minimum de 1. Cela signifie qu'une instance du concept « Parent » devrait au moins entretenir un rôle « enfant » avec une autre instance. Cet opérateur est noté de la manière suivante:

Forme concrète : *(ATLEAST n R)*, *n* est un entier supérieur à 0.

Forme abstraite : $\geq n R$

Exemple TF : *Parent ≐ (AND Humain ATLEAST 1 enfant)*

e) **ATMOST** : Cet opérateur permet d'effectuer une restriction de cardinalité maximum sur un rôle donné. Il sert à définir une borne supérieure sur un rôle. En utilisant cet opérateur et l'opérateur *ATLEAST*, il est possible de définir

complètement le domaine des valeurs permises pour un rôle. Par exemple, un concept « Personne » pourrait se voir imposer une cardinalité maximum de 2 sur un rôle nommé « géniteur ». Cet opérateur utilise la notation suivante :

Forme concrète : *(ATMOST n R)*, *n* est un entier supérieur à 0.

Forme abstraite : $\leq n\ R$

Exemple TF : *Personne* $\doteq$ *(AND Humain ATMOST 2 géniteur)*

3.3.2.3 Construction de rôles

Les rôles occupent une place particulièrement importante en logique terminologique. Ils servent à caractériser les concepts et permettent de définir les relations qu'ils entretiennent les uns avec les autres. Tout comme les concepts, il existe en langage TF trois façons de définir un rôle. La première consiste à définir un rôle de base qui n'est la spécialisation d'aucun autre rôle existant. Ce type de rôle devra être issu du rôle maximal T. Par exemple, on pourrait définir un rôle nommé « mère » comme étant issu de T. La seconde consiste à définir un rôle comme étant une spécialisation d'un autre rôle existant. Par conséquent, cette action définit également une relation de subsomption entre ces deux rôles. Par exemple, on pourrait définir un rôle « mère-célibataire » comme étant une spécialisation du rôle « mère ». La dernière façon consiste à créer un rôle qui est équivalent à un autre. En fait, il s'agit de donner un deuxième nom à un rôle qui existe déjà. On pourrait, par exemple, définir un rôle nommé « maman » et spécifier qu'il est l'équivalent du rôle « mère ». Une grande partie des langages terminologiques permettent également la conception de rôles complexes qui peuvent être introduits à l'aide de conjonctions de

plusieurs rôles ou via d'autres opérateurs. Cependant, le langage TF ne permet pas la création de rôles complexes. Il est tout de même suffisamment complet pour modéliser un grand nombre de situations [Nebel, 1995]. Voici les trois notations disponibles en langage TF pour l'introduction de nouveaux rôles :

<Rôle atomique> $\leq$ T

<Rôle atomique> $\leq$ *<Rôle atomique existant>*

<Rôle atomique> $\doteq$ *<Rôle atomique existant>*

Exemples TF :

mère $\leq$ T

mère-célibataire $\leq$ *mère*

maman $\doteq$ *mère*

La figure 3.1 nous montre la syntaxe complète du langage TF décrit par Nebel [Nebel, 1995]. Toutes les notions peuvent être décrites de façon compacte en utilisant le format BNF. Cette notation est semblable à celle utilisée pour la description des grammaires hors-contexte en théorie des automates. Il est à noter ici que le langage TF ne représente que la partie terminologique d'un formalisme de description; la partie assertionnelle associée au langage TF s'appelle le langage AF. Cette partie assertionnelle sert à l'introduction et à la manipulation d'individus issus des concepts et des rôles décrits par la partie terminologique. La conjonction des deux langages constitue un formalisme terminologique à part entière.

Syntaxe de TF

⟨ *terminologie* ⟩	::=	{ ⟨ *termeIntroduction* ⟩ | ⟨ *restriction* ⟩ }
⟨ *termeIntroduction* ⟩	::=	⟨ *conceptIntroduction* ⟩ | ⟨ *roleIntroduction* ⟩
⟨ *conceptIntroduction* ⟩	::=	⟨ *atomicConcept* ⟩ ≐ ⟨ *concept* ⟩ | ⟨ *atomicConcept* ⟩ ≤ ⟨ *concept* ⟩ | ⟨ *atomicConcept* ⟩ ≤ *ANYTHING*
⟨ *roleIntroduction* ⟩	::=	⟨ *atomicRole* ⟩ ≐ ⟨ *role* ⟩ | ⟨ *atomicRole* ⟩ ≤ ⟨ *role* ⟩ | ⟨ *atomicRole* ⟩ ≤ *ANYTHING*
⟨ *concept* ⟩	::=	⟨ *atomicConcept* ⟩ | *(AND* ⟨ *concept* ⟩$^{+}$ *)* | *(ALL* ⟨ *role* ⟩ ⟨ *concept* ⟩ *)* | *(ATLEAST* ⟨ *nombre* ⟩ ⟨ *role* ⟩ *)* | *(ATMOST* ⟨ *nombre* ⟩ ⟨ *role* ⟩ *)*
⟨ *role* ⟩	::=	⟨ *atomicRole* ⟩
⟨ *restriction* ⟩	::=	*(DISJOINT* ⟨ *atomicConcept* ⟩ ⟨ *atomicConcept* ⟩ *)*
⟨ *nombre* ⟩	::=	⟨ *entierNonNegatif* ⟩
⟨ *atomicRole* ⟩	::=	⟨ *identifieur* ⟩
⟨ *atomicConcept* ⟩	::=	⟨ *identifieur* ⟩

Fig. 3.1 : Syntaxe du langage TF (format BNF)

3.3.3 Sémantique du langage TF

À l'instar de la logique classique, une sémantique est associée aux descriptions de concepts et de rôle en logique terminologique. Les concepts sont interprétés comme des sous-ensembles d'un domaine d'interprétation et les rôles comme des sous-ensembles d'un produit entre deux domaines d'interprétations [Napoli, 1997].

3.3.3.1 Notion d'interprétation

La notion d'interprétation, symbolisée par *I*, est surtout décrite dans les articles de Baader et Hollunder [Baader et Hollunder, 1991] [Baader et al., 1991] et de Falquet

[Falquet, 2001]. Pour la définir, il faut se donner un *domaine sémantique* Δ^I, qui correspond à l'ensemble de tous les individus de la terminologie étudiée (pas seulement ceux qui sont effectivement existants), et une fonction $.^I$, dont le rôle est d'associer chacun des termes désignant un concept à un sous-ensemble de Δ^I et chaque nom de rôle à un sous-ensemble de $\Delta^I \times \Delta^I$. De plus, la fonction $.^I$ est appelée *fonction d'interprétation* et elle s'utilise en respectant la terminologie introduite par le langage terminologique sous-jacent. Prenons, par exemple, les concepts suivants : {Chaise, Dossier, Pied} et le rôle « partie » qui relie une composante d'un meuble à celui-ci. Le domaine sémantique Δ^I sera l'ensemble de tous les individus issus des concepts « Chaise », « Dossier » et « Pied » ainsi que toutes les parties possibles (respectivement c_i, d_i, p_i pour représenter les concepts et des couples de ceux-ci pour représenter le rôle « partie », en respectant sa définition). Donc, en appliquant une fonction d'interprétation, on aurait, par exemple :

$$Chaise^I = \{ c_1, c_2, c_3, c_4 \}$$
$$Dossier^I = \{ d_1, d_2, d_3 \}$$
$$Pied^I = \{ p_1, p_2, p_3, p_4 \}$$
$$partie^I = \{ (c_1, d_1), (c_1, p_1), (c_1, p_2), (c_1, p_3), (c_1, p_4), (c_3, d_2) \}$$

La notion d'interprétation signifie littéralement que l'on interprète les concepts et les rôles donnés par une terminologie. Il peut donc y avoir plusieurs interprétations I possibles pour une terminologie donnée.

3.3.3.2 Illustration ensembliste de TF

Il est possible de réécrire tous les opérateurs d'un langage terminologique avec un symbolisme ensembliste. C'est ce qu'on appelle la sémantique d'un langage terminologique. On distingue souvent une terminologie de son interprétation en utilisant les termes *définition par intention* et *définition par extension* pour l'interprétation. Voici, à titre d'exemple, la sémantique du langage TF, sous le format « *forme concrète* = *forme abstraite* = *sémantique* », où C et D désignent deux concepts :

(le « tout »)^I	=	T^I	=	Δ^I
(le « rien »)^I	=	$\bot^I$	=	$\varnothing^I$
(AND)^I	=	$(C \sqcap D)^I$	=	$C^I \cap D^I$
(ALL)^I	=	$(\forall\, R : C)^I$	=	$\{d \in \Delta^I \mid (d, e) \in R^I \rightarrow e \in C^I,\ \forall e\}$
(ATLEAST)^I	=	$(\geq n\, R)^I$	=	$\{d \in \Delta^I \mid \#^3\ \{ e \mid (d, e) \in R^I\} \geq n\}$
(ATMOST)^I	=	$(\leq n\, R)^I$	=	$\{d \in \Delta^I \mid \#\{ e \mid (d, e) \in R^I\} \leq n\}$

La définition d'un concept A comme étant un sous-concept de B, c'est-à-dire $A \leq B$, s'écrit sémantiquement avec le symbole de l'inclusion des ensembles $A \subseteq B$. Quant à la définition d'un concept A à l'aide du symbole $\doteq$, elle est équivalente à la création d'un nouvel ensemble $A \subseteq \Delta^I$. Ceci s'applique également aux rôles, mais en sachant qu'un rôle est une fonction dans $\Delta^I \times \Delta^I$. À partir de là, il est tout à fait possible de présenter graphiquement la sémantique d'un langage terminologique, puisqu'elle est uniquement

composée d'opérations sur des ensembles. Par conséquent, une autre façon de visualiser une logique terminologique est par l'utilisation des ensembles. Il suffit d'appliquer les opérations ensemblistes : l'intersection pour le *AND*, l'union pour le *OR* (non disponible en TF, mais il se retrouve dans plusieurs autres langages terminologiques), etc. Cette équivalence ensembliste s'applique également à la relation de subsomption. Par conséquent, une relation de subsomption entre un concept A et un concept B peut être présentée comme l'inclusion d'un ensemble A dans un ensemble B ou vice-versa.

3.3.4 La subsomption

La relation de subsomption permet d'organiser les concepts et les rôles par niveaux de généralité. Intuitivement, un concept A subsume un concept B si A (le subsumant) est plus général que B (le subsumé). Cette réalité peut être exprimée de trois différentes façons :

a) **Définition ensembliste en extension:** un concept A subsume un concept B si et seulement si l'ensemble des représentants dénoté par A contient nécessairement l'ensemble des représentants dénoté par B [Levesque et Brachman, 1987]. Dans cette définition, chaque concept est représenté par un ensemble et chacune des relations de subsomption est représentée par une inclusion ensembliste. Par exemple, le concept qui dénote les mammifères subsume celui dénotant les chats.

---

[3] Dans le cas présent, le symbole # désigne la cardinalité d'un ensemble.

b) **Définition ensembliste en intention**: un concept *A* subsume un concept *B* si toute entité représentée par *B* l'est aussi par *A*. Ainsi, chaque entité dont la description est définie par *B* possède les caractéristiques qui sont spécifiées par *A* [Finin, 1986]. Par exemple, le concept dénotant les chats possède toutes les caractéristiques de celui qui dénote les mammifères en plus de celles qui sont propres aux chats. Un concept se compose donc d'une description propre à lui-même et d'une description « partagée » ou « commune » avec ses subsumants.

c) **Définition logique**: un concept *A* subsume un concept *B* si un élément qui est décrit par *B* implique logiquement que cet élément est décrit par *A*. Ainsi, l'implication *(Chat(x)* $\Rightarrow$ *Mammifère(x))* est vraie pour tout *x* [McGregor, 1988].

En utilisant la notion d'interprétation, on peut définir la relation de subsomption de la manière suivante : un concept *B* est subsumé par un concept *A* (respectivement *A* subsume *B*), ce qui se note $B \subseteq A$, si et seulement si $B^I \subseteq A^I$ pour toute interprétation *I*.

La subsomption est à la base de la classification dans la logique terminologique. C'est une relation d'ordre partiel qui permet d'organiser les concepts en une hiérarchie. Chaque concept possède une description qui permet de le comparer aux autres et ainsi de déterminer les relations de subsomption qu'il entretient avec les autres concepts. La subsomption est une relation réflexive; un concept *A* est donc subsumé par lui-même. Cette relation est également transitive, ce qui signifie que si un concept *A* subsume un concept *B*

et que ce concept B subsume un concept C, alors le concept A subsume automatiquement le concept C. La relation de subsomption est aussi antisymétrique, ce qui signifie que si un concept A subsume un concept B et que le concept B subsume à son tour le concept A, alors $A = B$. Les différentes relations de subsomption qu'entretient un concept permettront de déduire l'emplacement que ce concept devrait occuper dans la hiérarchie. Un concept maximal, représentant la totalité du domaine d'interprétation, est présent au sommet de la hiérarchie. Ce concept représente l'ensemble le plus général et subsume tous les autres, permettant ainsi d'obtenir un point de départ pour la taxonomie. Inversement, un concept minimal est présent au bas de la hiérarchie. Ce concept est subsumé par tous les autres; il représente en fait le vide ou le « rien ».

Prenons, par exemple, les concepts « Animal », « Mammifère », « Ovipare », « Reptile » et « Ornithorynque », tels que représentés de façon ensembliste sur la figure 3.2. On peut voir que le concept « Ovipare » subsume le concept « Reptile », car le fait d'être un « Reptile » implique le fait d'être un « Ovipare ». On peut également noter, sur la même figure, que le concept « Ornithorynque » est subsumé par le concept « Mammifère » et par le concept « Ovipare », mais de façon plus explicite, par la conjonction des deux concepts. Le fait d'être un « Ornithorynque » implique obligatoirement le fait d'être un « Mammifère » et un « Ovipare ». Chacun des concepts est donc délimité par l'ensemble des individus qu'il représente par rapport aux autres concepts. On peut donc en conclure que la classification d'un nouveau concept se réduit à déterminer où devra se situer l'ensemble qu'il représente par rapport aux autres ensembles déjà existants.

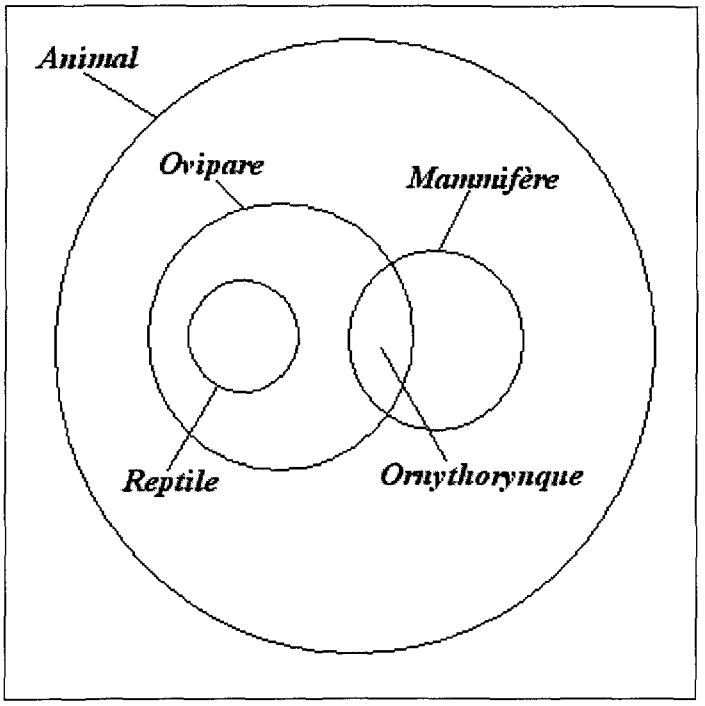

Fig. 3.2 : Représentation ensembliste d'une série de concepts

3.3.4.1 Subsomption versus Héritage

Il est très important de faire la distinction entre une relation d'héritage (telle que l'on connaît dans les langages objets) et une relation de subsomption. À première vue, les deux relations peuvent sembler identiques, mais même si elles s'apparentent l'une à l'autre, elles ne le sont pas. En vérité, on peut dire que la relation de subsomption subsume ou est plus générale que la relation d'héritage, car elle porte sur les concepts, sur les individus, mais aussi, de façon duale, sur des clauses : les concepts s'apparentent à des conjonctions de littéraux de la logique du premier ordre [Hamburger et Richards, 2002] alors que les clauses sont des disjonctions de littéraux [Borgida, 1996]. La relation de subsomption relève donc d'un niveau d'abstraction plus élevé que la relation d'héritage. On peut voir

l'héritage comme un mécanisme de partage de la connaissance qui peut être considéré comme une façon d'implanter la subsomption [Haton et al., 1991]. De plus, l'héritage est une relation tout ou rien, ce qui n'est pas le cas de la subsomption.

3.3.4.2 Détection des relations de subsomption

Afin de pouvoir identifier une relation de subsomption entre deux concepts, il faut les comparer « composante par composante ». Il faut se rappeler qu'un concept en logique terminologique peut être vu comme une conjonction de caractéristiques définies par des rôles et que des restrictions de cardinalités peuvent être attribuées à ces caractéristiques. Autrement dit, pour vérifier si un concept A subsume un concept B, il faut s'assurer que pour chaque rôle du concept B, il existe un rôle équivalent ou plus général défini par le concept A. Afin de mieux comprendre la méthode de détection, voici une version simplifiée de l'algorithme utilisé avec KL-ONE par Schmolze et Lipkis [Schmolze et Lipkis, 1983] pour déterminer si un concept A subsume un concept B :

Si tous les concepts qui sont subsumants de A sont aussi subsumants de B
    Alors retourner vrai
Pour tous les rôles a_i de A
    Pour tous les rôles b_j de B
        Si l'un des rôles b_j de B exprime le même rôle que a_i de A (si $a_i = b_j$) OU
        Si l'extension du rôle a_i est incluse par l'extension du rôle b_j (si a_i subsume b_j)
            Alors passer à (a_{i+1})
        Sinon retourner échec ou inconnu
Retourner vrai

Il faut maintenant préciser comment il est possible de déterminer si un rôle a_i subsume un rôle b_j. Pour cela, il faut répondre aux deux questions suivantes :

- Est-ce que le co-domaine du rôle a_i subsume le co-domaine du rôle b_j?
- Est-ce que la cardinalité du rôle a_i est égale ou supérieure à celle du rôle b_j?

Si les réponses aux questions précédentes sont vraies, alors on peut affirmer que le rôle a_i subsume le rôle b_j.

L'algorithme de vérification d'une relation de subsomption présenté ci-haut peut retourner deux réponses : vrai ou inconnu. Dans le cas où l'algorithme retourne vrai, cela signifie que le concept A subsume avec certitude le concept B. Cependant, lorsque l'algorithme retourne inconnu, cela ne signifie pas nécessairement que le concept A ne subsume pas le concept B, mais plutôt qu'avec les informations connues sur les deux concepts, il est impossible de le prouver. Le problème de la détection des relations de subsomption est d'une importance capitale pour la classification dans une hiérarchie de concepts.

3.3.5 Les différents formalismes terminologiques

Il existe un nombre impressionnant de formalismes terminologiques; le langage TF utilisé dans la section précédente n'est qu'un exemple parmi tant d'autres. Les principales différences entre les langages terminologiques résident dans le nombre d'opérateurs

disponibles pour la construction de concepts et de rôles, ainsi que dans la syntaxe utilisée par ceux-ci. Le premier formalisme à avoir vu le jour fut le langage FL^-, conçu par Brachman et Schmolze [Brachman et Schmolze, 1985] pour KL-ONE. Ce langage ne comportait que trois opérateurs, soit *AND*, *ALL* et *SOME*. Il a donné naissance à la famille des langages FL. Tous les langages de cette famille utilisent la syntaxe issue de FL^- en y ajoutant certains opérateurs. Il est à noter que le langage TF utilisé précédemment est basé, lui aussi, sur le langage FL^-.

Dans la littérature, la famille de langages terminologiques de références est la famille AL. La particularité de cette famille est que chaque langage qui en fait partie comporte la syntaxe et les opérateurs de base AL, plus un certain nombre d'opérateurs. Chaque opérateur additionnel est représenté par une lettre qui est ajoutée au nom du langage. Par exemple, le langage ALUC est composé des opérateurs de base de AL, ainsi que de l'opérateur de disjonction, représenté par la lettre U, et l'opérateur de négation, représenté par la lettre C. Les langages AL forment la plus grande famille de langages terminologiques. La formule générale pour la formation des langages AL est la suivante :

AL[E] [U] [C] [N] [O] [B]

Le tableau 3.1 présente une description de la majorité des opérateurs terminologiques.

Légende

C et D	=	Désignent deux concepts
R et S	=	Désignent deux rôles
a et b	=	Désignent deux individus spécifiques
n	=	Désigne un entier positif

Nom	Lettre (AL)	Symb. concret	Symb. abstrait	Symb. Sémantique	Commentaire
Tout	-	TOP	$\top$	Δ^I	-
Rien	-	BOTTOM	$\bot$	$\varnothing$	-
Conjonction	-	AND	$C \sqcap D$	$C^I \cap D^I$	-
Disjonction	U	OR	$C \sqcup D$	$C^I \cup D^I$	-
Complément	C	NOT	$\neg C$	$\Delta^I \setminus C^I$	-
Quantification universelle	-	ALL	$\forall R C$	-	-
Quantification existentielle non qualifiée	-	SOME	$\exists R$	-	Il existe un rôle R sans restriction sur le co-domaine
Quantification existentielle qualifiée	E	SOME	$\exists R C$	-	Il existe un rôle R qui doit relier un concept C
Restriction de nombre	N	ATLEAST ATMOST EXACT	$\geq n R$ $\leq n R$ $n R$	-	S'applique sur le nombre de rôles de type R
Égalité	-	EQ	$R = S$	$R^I = S^I$	S'applique sur les rôles et les concepts
Inégalité	-	NEQ	$R \neq S$	$R^I \neq S^I$	S'applique sur les rôles et les concepts
Sous-ensemble	-	SUBSET	$R \subseteq S$	$R^I \subseteq S^I$	S'applique sur les rôles et les concepts
Collection	O	ONEOF	$\{ a, b, \ldots \}$	$\{ a^I, b^I, \ldots \}$	Précise la liste des individus qui sont des instances d'un concept
Contrainte d'individus	B	IS	$R : a$	-	Restreint un rôle R à être relié à un individu a
Attributs	F	-	-	-	Permet l'utilisation d'attributs

Tableau 3.1 : Liste des opérateurs communs des langages terminologiques

En plus des deux grandes familles que forment FL et AL, il existe certains langages qui n'appartiennent à aucune de ces familles. Plusieurs SRCT possèdent leur propre langage terminologique adapté à leurs besoins. Par exemple, les langages terminologiques utilisés par les SRCT CLASSIC [Resnick et al., 1995] et LOOM [LOOM, 1991]. D'autres langages ayant eu un impact moindre ont également été développés. Par exemple, le langage U, développé par Patel-Schneider en 1987 [Schmidt, 1991][Nebel, 1995], qui se voulait un langage terminologique universel.

3.3.6 Comparaison de la logique terminologique avec la logique du premier ordre (LPO)

La logique terminologique et la logique du premier ordre sont très proches l'une de l'autre. Par ailleurs, les concepts que l'on retrouve en logique terminologique peuvent être vus comme des prédicats unaires, et les rôles comme des prédicats binaires. Il est toujours possible d'effectuer la conversion d'une réalité modélisée avec un langage terminologique en LPO, mais l'inverse n'est pas toujours vrai. On peut donc en conclure que la LPO subsume la logique terminologique. Cependant, la LPO est une logique mathématique difficilement utilisable de façon concrète pour la classification, contrairement à la logique terminologique.

Une comparaison entre le pouvoir d'expression de la logique terminologique et celui de la logique du premier ordre est donnée par Baader et Borgida [Baader, 1999] [Borgida, 1996]. Leurs comparaisons établissent que tous les langages terminologiques

proposés jusqu'à présent peuvent exprimer *toutes* les notions exprimables dans la LPO avec trois variables *au plus*, et *seulement* celles-là [Borgida, 1996].

3.4 Les systèmes de représentation de la connaissance terminologique (SRCT)

Jusqu'à maintenant, il a surtout été question de la partie terminologique de la logique terminologique, qui sert principalement à définir des concepts et des rôles de façon à pouvoir créer un univers fini où l'on peut classifier les concepts. Dans cette section du chapitre, nous allons nous attarder plus en détails sur la partie assertionnelle de la logique terminologique. Cette partie assertionnelle nous permet de créer des individus issus des concepts, de manière à peupler notre univers fini.

3.4.1 Les composantes d'un SRCT

Un SRCT est un système terminologique permettant l'implantation d'une base de connaissances. Il constitue une application, dans un contexte réel, de la théorie sur la logique terminologique. Tous les SRCT sont composés de trois parties bien distinctes, comme nous le montre la figure 3.3 : la partie terminologique (TBox) qui permet de définir un univers fini, la partie assertionnelle (ABox) qui permet l'introduction d'individus dans cet univers et un moteur d'inférence qui utilise la ABox et la TBox pour fournir différents services aux utilisateurs.

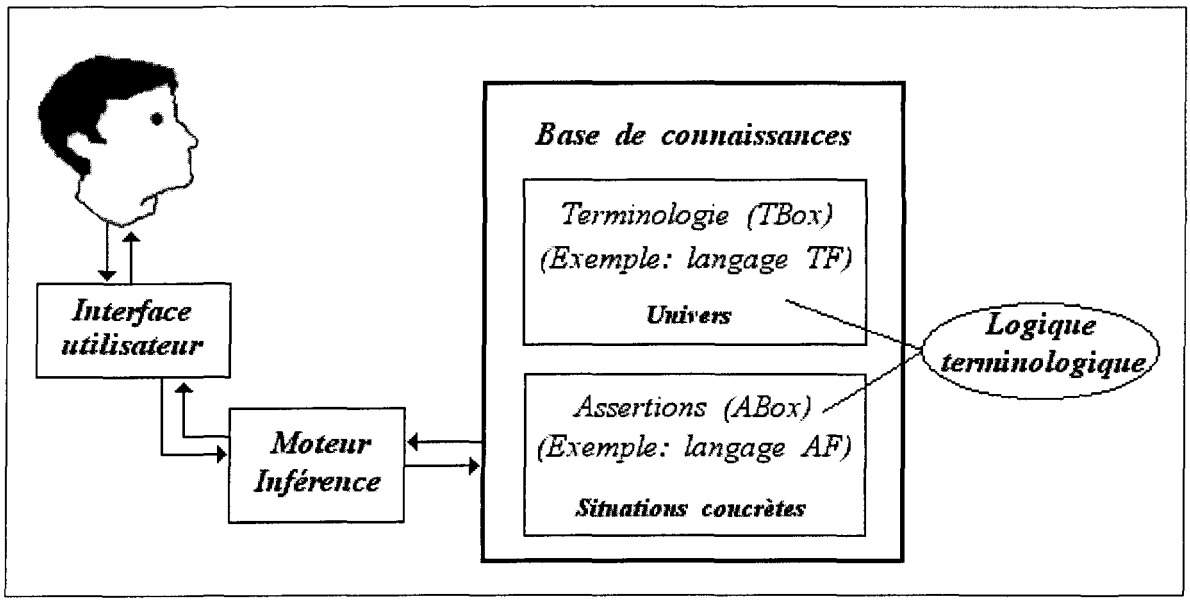

Fig. 3.3 : Schéma général d'un SRCT

La TBox est essentiellement composée d'une série de définitions terminologiques. Elle est décrite par un langage terminologique précis, par exemple le langage TF. Les définitions introduites dans une TBox sont habituellement non cycliques. Cependant, il est possible de créer un SRCT qui supporte certaines définitions cycliques, comme la récursion et les structures d'objets infinis [Nebel, 1995].

En ce qui concerne la ABox, elle est constituée d'une série d'assertions d'individus, écrites à l'aide de la syntaxe assertionnelle correspondant au langage terminologique utilisé par la TBox. Par exemple, si la TBox utilise la syntaxe du langage TF, la ABox devra obligatoirement utiliser celle du langage AF, qui est son complément assertionnel. La section suivante présentera plus en détails le langage assertionnel AF.

Finalement, le moteur d'inférence est constitué de fonctions et d'algorithmes permettant de fournir des services aux utilisateurs. Il permet de déduire de nouveaux faits en fonction de ceux définis dans la ABox, en se fiant aux règles de l'univers contenues dans la TBox. Les services d'inférence fournis par les différents SRCT seront décrit dans section 3.4.5 du chapitre.

3.4.2 Le langage assertionnel AF

Le langage assertionnel AF sert à introduire des individus réels représentant des instances de concepts ou de rôles, définis préalablement avec le langage terminologique TF. Chaque langage terminologique possède son complément assertionnel qui permet la création d'individus. Il est impossible d'utiliser un langage assertionnel seul, il doit toujours être utilisé avec son équivalent terminologique. La figure 3.4 nous présente la syntaxe complète du langage AF ; l'utilisation de cette syntaxe sera décrite dans les prochains paragraphes.

Syntaxe AF

⟨*description*⟩	::=	(⟨*objetDescription*⟩ | ⟨*roleDescription*⟩)
⟨*objetDescription*⟩	::=	(⟨*atomicConcept*⟩ ⟨*objet*⟩)
⟨*roleDescription*⟩	::=	(⟨*atomicRole*⟩ ⟨*objet*⟩ ⟨*objet*⟩) | (⟨*atomicRole*⟩ ⟨*objet*⟩ (*ATLEAST* ⟨*nombre*⟩)) | (⟨*atomicRole*⟩ ⟨*objet*⟩ (*ATMOST* ⟨*nombre*⟩))

Fig. 3.4 : Syntaxe du langage assertionnel AF

Le langage assertionnel AF permet principalement trois choses : l'introduction d'individus issus de concepts, l'introduction d'instances de rôles et l'attribution de restrictions propres aux individus. Il est à noter que dans les exemples qui suivront, le symbole « # » précèdera chaque nom d'individus afin de les différencier des noms de concepts. Les notations suivantes nous permettront d'introduire des individus à l'aide du langage AF. Pour chacune des définitions, nous considèrerons *C* et *D* comme étant deux concepts, *a* et *b* comme des instances de ces concepts, et *R* comme étant un rôle:

a) **Introduction d'un individu issu d'un concept :** cette opération permet de définir des individus spécifiques issus de concepts existants. On peut, par exemple, introduire l'individu nommé « #John » qui est issu du concept « Humain ».

Notation concrète	:	*(C a)*
Notation concrète détaillée	:	*(Concept nomConcept)*
Notation abstraite	:	*C(a)*
Exemple en AF	:	*(Humain #John)*

b) **Introduction d'une instance de rôle :** cette opération permet de définir qu'un rôle donné existe effectivement entre deux individus. Par exemple, si un rôle nommé « marié » existe dans la terminologie comme étant une relation entre deux « Humains », on peut définir que l'individu « #John » est marié à l'individu « #Claire ».

Notation concrète	:	*(R a b)*
Notation concrète détaillée	:	*(Rôle nomIndividu1 nomIndividu2)*

Notation abstraite	:	*R(a, b)*
Exemple en AF	:	*(marié #John #Claire)*

c) **Introduction de restrictions sur le rôle d'un individu :** cette opération permet de définir des restrictions de cardinalité spécifique à un individu. Par exemple, supposons qu'un concept nommé « Équipe » soit défini comme ayant au plus 4 « membre », mais qu'un individu spécifique, par exemple « #ÉquipeA », ne puisse être composé que de trois membres au plus. Il sera possible de définir cette particularité à l'aide du langage AF.

Notation concrète	:	*(R a (ATLEAST n)), où $n \geq 1$*
	:	*(R a (ATMOST n))*
Notation concrète détaillée	:	*(Rôle nomIndividu (ATLEAST n))*
	:	*(Rôle nomIndividu (ATMOST n))*
Notation abstraite	:	*R(a, ($\geq$ n))*
	:	*R(a,($\leq$ n))*
Exemple en AF	:	*(membre #ÉquipeA (ATMOST 3))*

3.4.3 La notion de modèle dans un SRCT

Afin d'être en mesure de bien comprendre les mécanismes et les services offerts par les SRCT, il faut revenir sur la notion d'interprétation que nous avons introduite à la section 3.3.3.1, car cette notion s'applique également sur les langages assertionnels (par exemple AF). Par conséquent, l'interprétation d'une instance d'un concept (soit a^I) ou d'un rôle (soit

$(a, b)^I$) peut être vue comme étant un individu précis de l'ensemble déterminé par le concept ou le rôle en question. L'exemple de la figure 3.5 illustre, à l'aide du domaine sémantique des êtres humains, cette notion d'interprétation des instances.

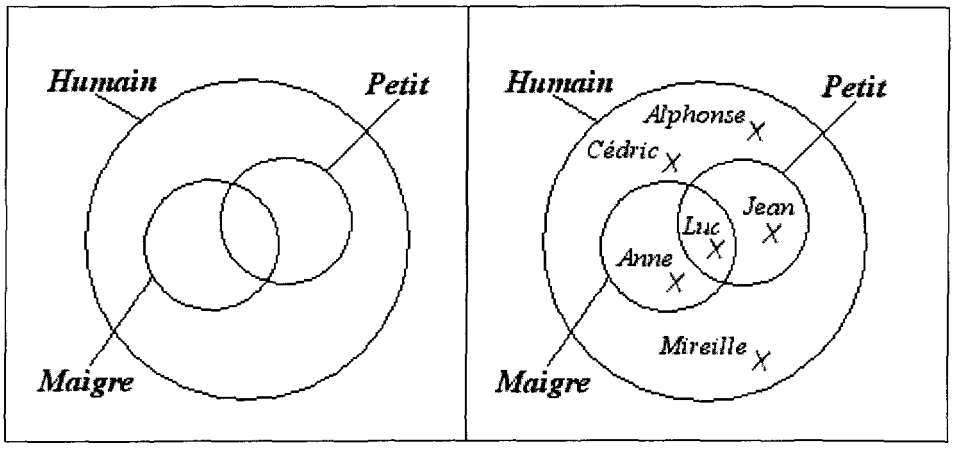

Fig. 3.5 : Introduction d'individus dans une terminologie de concepts

Pour les prochaines définitions, nous allons considérer les variables suivantes :

I	:	*une interprétation.*
C et D	:	*deux concepts.*
R	:	*un rôle.*
a et b	:	*deux individus (instances) de C et D.*

Une interprétation *I satisfera une TBox T* si et seulement si, pour toutes les définitions terminologiques incluses dans *T*, les deux conditions suivantes sont vérifiées et valides :

a) Pour toutes les définitions de concepts avec le symbole $\doteq$, c'est-à-dire $C \doteq D$ (où D peut être complexe), il est possible de vérifier que $C^I = D^I$.

b) Pour toutes les définitions de concepts avec le symbole $\preceq$, c'est-à-dire $C \preceq D$, il est possible de vérifier que $C^I \subseteq D^I$.

Ces conditions signifient qu'il existe un ou plusieurs sous-ensembles de Δ^I satisfaisant toutes les définitions terminologiques de T. Lorsque cela se produit, l'interprétation I devient un *modèle* de la TBox en question. Dans ce cas, la TBox est dite *cohérente*.

La même définition s'applique pour une ABox. Par conséquent, une interprétation I *satisfera une ABox A* si et seulement si, pour toutes les définitions assertionnelles incluses dans A, les deux conditions suivantes sont remplies :

a) Pour toutes les instanciations de concepts de la forme $C(a)$, il est possible de vérifier que $a^I \in C^I$. Autrement dit, cela signifie que les individus créés par le langage assertionnel existent vraiment dans les ensembles représentant les concepts (atomiques ou complexes) en question.

b) Pour toutes les instanciations de rôles de la forme $R(a, b)$, il est possible de vérifier que $R(a^I, b^I) \in R^I$. Autrement dit, les relations créées par le langage assertionnel

existent vraiment dans les ensembles représentant les rôles (atomiques ou complexes).

Dans le cas présent, la signification des conditions est basée sur l'existence réelle des individus dans l'univers représenté. Par exemple, si les instances créées par le langage assertionnel ne sont pas contenues dans le domaine sémantique Δ^I, alors, ces conditions seront automatiquement invalides. Comme dans le cas de la TBox, une interprétation qui satisfait une ABox va s'appeler un *modèle* de la ABox.

Si une interprétation I est à la fois un modèle de la TBox et de la ABox d'un SRCT donné, alors cette interprétation est considérée comme un *modèle* du SRCT. Sachant cela, il sera plus facile de comprendre les services offerts par les SRCT ainsi que les langages assertionnels.

3.4.4 Raisonnement dans un SRCT

Afin de pouvoir raisonner et exploiter les connaissances de la TBox et de la ABox, un SRCT offre un certain nombre de services tels que l'héritage, la restriction et la propagation de restrictions, la détection d'incohérences sur les descriptions de concepts et la classification. Une description des différents services et mécanismes qui sont supportés par les SRCT est présentée dans les articles de Baader [Baader et Hollunder, 1991], Valancia [Valancia, 2000] et Nebel [Nebel, 1995]. À titre d'illustration, nous décrirons le mécanisme de classification qui nous intéresse pour la compréhension du chapitre 4.

3.4.4.1 La classification

La classification permet de déterminer la position d'un concept ou d'un rôle dans la hiérarchie. L'emplacement de ceux-ci est d'une importance capitale lors des opérations d'inférence. Ce service utilise les relations de subsomption décrites à la section 3.3.2, afin de déterminer le positionnement des différents concepts et rôles. Il entre en fonction chaque fois qu'un nouveau concept ou un nouveau rôle est défini dans la terminologie (TBox). Le processus de classification d'un nouveau concept se décompose en deux étapes : la recherche des subsumants les plus spécifiques (SPS) et la recherche des subsumés les plus généraux (SPG). Une fois ces deux étapes complétées, il suffit d'ajouter les relations nécessaires (par exemple une relation d'héritage) entre le nouveau concept et les différents SPS et SPG trouvés pour positionner ce concept dans la hiérarchie.

L'exemple suivant nous montre un algorithme de classification inspiré de celui utilisé par KL-ONE [Lipkis, 1982]. La recherche des subsumants d'un concept à classer X commence au sommet de la hiérarchie et se poursuit en profondeur. La comparaison du concept courant O^* avec le concept à classer X s'établit de la façon suivante :

COMPARER(O^*, X)
  Si O^* ne subsume pas X
  Alors ***(la hiérarchie est bien formée : aucun descendant de O^* ne peut subsumer X)***
    Retourner nil
  Sinon ***(O^* est temporairement le subsumant le plus spécifique)***
    Si O^* est une feuille dans la hiérarchie
    Alors
      retourner { O^* }

```
Sinon (SPS1 est une variable locale)
    SPS1 = nil
    Pour chaque descendant D* de O* Faire
        SPS1 = SPS1 ∪ COMPARER( D*, X )
        (Si SPS1 a une valeur,
        alors un des descendants de O* est le subsumant le plus spécifique)
        Si SPS1 = nil
        Alors
            retourner { O* }
        Sinon
            retourner SPS1
```

Seule une partie de la hiérarchie est explorée : si O^* ne subsume pas X, alors le sous-arbre d'héritage de la racine O^* est élagué. L'exploration se poursuit à partir du premier frère de O^* faisant encore partie de l'ensemble des concepts à explorer. Si O^* subsume X, alors O^* est temporairement le subsumant le plus spécifique du sous-arbre courant. L'exploration se poursuit en profondeur à partir de O^*. Si aucun des descendants de O^* ne subsume X, alors O^* est déclaré être le subsumant le plus spécifique du sous-arbre courant. Une nouvelle relation, un lien d'héritage par exemple, peut-être mise en place entre le concept X et ses subsumants les plus spécifiques contenus dans SPS.

La seconde partie de l'algorithme concerne la recherche des subsumés les plus généraux. L'obtention des subsumants les plus spécifiques permet de focaliser cette recherche : puisque la relation de subsomption est transitive, il suffit de n'explorer que les descendants des SPS qui possèdent les mêmes propriétés que X. Si $\{ D \}$ désigne l'ensemble des descendants des SPS et D^* le descendant courant, l'exploration s'effectue

sur { D } de la façon suivante : si X subsume D^*, alors ce dernier est un SPG, sinon l'exploration se poursuit sur les descendants de D^* jusqu'à ce qu'un SPG ait été trouvé ou bien que D^* n'ait pas de descendant.

3.4.5 LOOM

Le SRCT appelé LOOM, conçu par l'Information Science Institute (ISI) de l'*University of Southern California*, se veut un langage très riche et complet, ce qui en fait un langage très complexe intégrant une multitude de fonctionnalités [MacGregor, 1991][MacGregor et Burstein, 1991]. LOOM est l'un des SRCT possédant le plus d'opérateurs de définitions pour les concepts et les rôles. Une grammaire simplifiée de LOOM est présentée à la figure 3.6.

Les constructeurs qui sont inclus dans la majorité des langages (à l'exception du constructeur *TEST* qui est unique à CLASSIC) sont supportés par la grammaire de LOOM. Il faut aussi noter que les constructeurs *OR* et *NOT*, qui ont été écartés en CLASSIC, ainsi que dans plusieurs autres langages, sont également supportés. Soulignons également l'existence du constructeur « exactly », qui est propre à LOOM. Ce constructeur spécial permet de définir une restriction de cardinalité sur un rôle de façon exacte. Par exemple, on pourrait définir que le rôle « soutenu-par », qui implique un concept « Chaise » et un concept « Pied », doit être de cardinalité 4. De cette façon, on peut spécifier qu'une « Chaise » a exactement 4 « Pied ». Il est à noter qu'il est possible de représenter cette même situation, dans les autres formalismes terminologiques, avec la conjonction d'une

restriction *AT-MOST* avec une restriction *AT-LEAST*. Par exemple, on peut définir le rôle « soutenu-par » comme étant de cardinalité minimum 4 (avec *AT-LEAST*), et maximum 4 (avec *AT-MOST*). Finalement, les constructeurs « domain » et « range » permettent de spécifier le domaine et le co-domaine d'un rôle, ce qui ajoute à la richesse du langage.

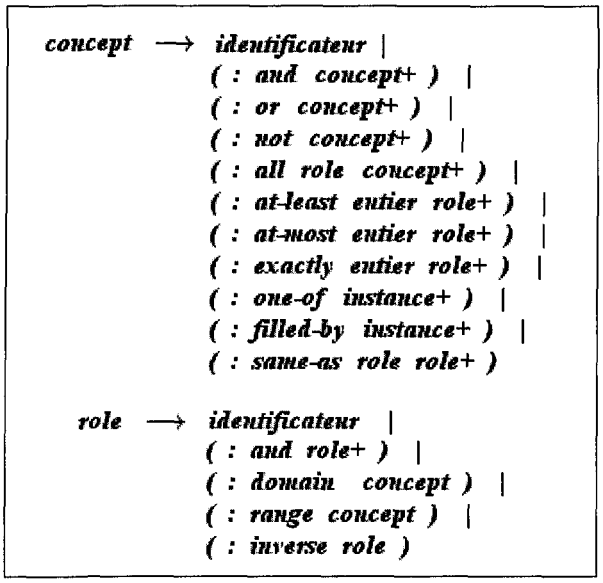

Fig 3.6 : Une grammaire simplifiée du langage de LOOM [LOOM, 1991].

3.4.5.1 Comparaison des formalismes LOOM et TF

LOOM possède un formalisme propre à lui-même. Pour illustrer cette particularité, l'exemple de la figure 3.7 nous montre la représentation d'une situation réelle en langage TF et la compare avec la même modélisation effectuée avec le formalisme de LOOM. Il faut bien remarquer le niveau de détails que l'on peut obtenir avec LOOM, grâce aux

définitions de concepts et de rôles. On peut noter, en regardant la figure 3.7, que les déclarations de concepts et de rôles, qui s'effectuaient en TF à l'aide des symboles $\doteq$ et $\leq$, s'effectuent ici grâce aux commandes « defconcept » et « defrelation ».

TF	*LOOM*
Être ≤ *T* *Ensemble* ≤ *T* *Humain* ≤ *Être* *Homme* ≤ *Humain* *Femme* ≤ *Humain* *(disjoint Homme Femme)* *Robot* ≤ *Être* *(disjoint Robot Humain)* *membre* ≤ *T* *chef* ≤ *membre* *Équipe* ≐ *(AND Ensemble (ALL membre Être) (ATLEAST 2 membre))* *Petite-Équipe* ≐ *(AND Équipe (ATMOST 5 membre))* *Équipe-Moderne* ≐ *(AND Équipe (ATMOST 4 membre) (ATLEAST 1 chef) (ALL chef ROBOT))*	*(defconcept ETRE :is :primitive)* *(defconcept ENSEMBLE :is :primitive)* *(defconcept HUMAIN :is (:and ETRE :primitive))* *(defconcept HOMME :is (:and HUMAIN :primitive))* *(defconcept FEMME :is (:and HUMAIN (:not HOMME) :primitive))* *(defconcept ROBOT :is (:and ETRE (:not HUMAIN) :primitive))* *(defrelation membre :is :primitive))* *(defrelation chef :is (:and membre :primitive))* *(defconcept EQUIPE :is (:and ENSEMBLE (:all membre ETRE) (:at-least 2 membre)))* *(defconcept PETITE-EQUIPE :is (:and EQUIPE (:at-most 5 membre)))* *(defconcept EQUIPE-MODERNE :is (:and EQUIPE (:at-most 4 membre) (:at-least 1 chef) (:all chef ROBOT))*

Fig. 3.7 : Comparaison du langage TF avec le formalisme de LOOM

Une autre particularité vient du fait que les notions de concepts définis et primitifs apparaissent de façon explicite dans la définition des concepts, via la commande « :primitive », utilisée lors de la définition des concepts « ETRE », « ENSEMBLE », « HUMAIN », « HOMME », « FEMME » et « ROBOT ». L'absence de la commande « :primitive » lors de la définition des concepts « EQUIPE », « PETITE-EQUIPE » et

« EQUIPE-MODERNE » spécifie implicitement que ces concepts sont bien définis. Finalement, il faut noter que les disjonctions, définies en TF par l'opérateur « disjoint », sont effectuées ici à l'aide de l'opérateur de négation « not ».

3.4.5.2 Conclusion sur LOOM

LOOM se veut, à l'heure actuelle, l'un des formalismes de représentation terminologique les plus complets et les plus polyvalents. L'environnement de LOOM a été enrichi, notamment avec un système à base de règles de production [Yen *et al.*, 1991a][Yen et al., 1991b]. De plus, le système de raisonnement de LOOM utilise un moteur d'inférence très complet. C'est pour toutes ces raisons que LOOM est considéré comme l'un des SRCT les plus expressifs et les plus riches.

Plusieurs systèmes qui utilisent LOOM comme système de représentation de la connaissance ont été construits. Notamment, un système de gestion d'informations hétérogènes [Arens *et al.*, 1993], un système de raisonnement à partir de cas dans le domaine légal [Ashley et Aleven, 1994] ainsi qu'un système de bases de données hiérarchiques réparties [Chu *et al.*, 1993]. Il faut aussi mentionner la création d'une nouvelle version de LOOM appelée PowerLoom [PowerLoom, 1997] par ISI, que nous avons d'ailleurs utilisée comme plate-forme pour le développement de notre système de comparaison de points de vue. Il sera question de cette nouvelle version de LOOM dans le chapitre 4.

3.5 Conclusion

Ce chapitre a permis d'introduire la notion de logique terminologique et de système de représentation de la connaissance terminologique (SRCT). Cette introduction a été réalisée grâce à une définition de la notion de connaissance, suivie d'une présentation théorique des éléments de base constituant la logique terminologique, pour finalement terminer avec une présentation détaillée sur les SRCT. La plupart des explications ont été appuyées par des exemples concrets, dans le but de comprendre plus facilement la théorie. Les exemples ont notamment été illustrés à l'aide du formalisme terminologique TF et du langage assertionnel AF.

La logique terminologique possède plusieurs avantages en tant que formalisme de représentation de la connaissance. Elle est reconnue pour être particulièrement efficace pour le raisonnement par classification. Cette logique possède une sémantique bien définie. Ceci implique que l'existence des instances n'est pas donnée de façon opérationnelle ou par l'implémentation elle-même. Ce serait l'inverse si on affirmait que l'assertion « John est un père » était valide uniquement parce que le système a répondu « père » à la question « Qu'est-ce que John ». L'existence des instances est plutôt donnée par des modèles et des descriptions qui permettent à la logique terminologique de s'articuler dans un univers bien défini, et non d'être une simple suite de définitions statiques de références. Il faut également souligner que la logique terminologique est une logique formelle qui est beaucoup simple et intuitive d'utilisation que la logique du premier ordre, entre autres, parce qu'il n'y a pas de quantificateur.

De plus, il a été prouvé que la logique terminologique est complète et correcte [Baader et al., 1991], ce qui permet de l'utiliser dans plusieurs domaines connexes à l'intelligence artificielle. Par exemple, les travaux de Schmidt [Schmidt, 1992] [Schmidt, 2000] ont permis de voir la logique terminologique comme étant une solution possible pour la compréhension du langage naturel. On peut également voir les applications de la logique terminologique dans le domaine des systèmes multi-agents, pour tenter de résoudre le problème de l'hétérogénéité sémantique [Valancia, 2000] concernant l'utilisation de terme différent pour décrire une même réalité, ainsi que dans nos propres travaux, pour mesurer la similarité des points de vue entre un usager et son agent [Bouzouane et Bouchard, 2003]. La logique terminologique a aussi été proposée pour la modélisation de bases de données [Borgida, 1995]. Récemment, le consortium W3C a émis un guide pour tenter de standardiser les ontologies utilisées par les services Web avec le Web Ontology Language [McGuinness et al., 2002].

Malgré tous les avantages de la logique terminologique, son principal problème est sa dualité expressivité versus complexité [Nebel, 1995]. Autrement dit, plus un langage terminologique permet de modéliser un grand nombre de réalités, plus sa complexité théorique augmente au niveau des opérations d'inférences, en particulier pour la détection des relations de subsomption [Schmidt, 1991].

Finalement, bien que ce chapitre constitue une introduction détaillée à la logique terminologique, il faut savoir qu'il existe plusieurs points qui n'ont pas été présentés en

profondeur, notamment les mécanismes pour supporter les définitions cycliques et récursives, les différents problèmes reliés à la complexité des SRCT et le fonctionnement de certains services. Ce chapitre se veut cependant une introduction suffisamment complète pour comprendre la suite de ce document.

CHAPITRE 4

MODÈLE DE LA SIMILARITÉ DES POINTS DE VUE ET VALIDATION EN E-COMMERCE

4.1 Introduction

La coopération entre un usager et un agent artificiel agissant en son nom - coopération usager-agent - a toujours été considérée comme une problématique très complexe. Elle nécessite une approche particulière afin d'intégrer l'utilisateur dans la même boucle de réalisation d'une tâche conjointe [Dautenhahn, 2001]. En fait, cette cohabitation hybride est à la fois riche et complexe, du fait que l'usager et l'agent sont amenés non seulement à s'accorder sur les différents niveaux de réalisation de la tâche à l'intérieur d'une même boucle de résolution, mais aussi à gérer l'initiative mixte « qui contrôle qui? ». Afin d'assurer une coopération usager-agent efficace, il est donc nécessaire pour eux d'être aptes à se comprendre mutuellement [Falcone et Castelfranchi, 2001]. Pour cela, il faut que ces deux entités soient capables de comparer leurs points de vue respectifs d'une même situation. Par ailleurs, dans les situations où une erreur inacceptable (voire dramatique) se produit due à une divergence de points de vue entre l'usager et son agent, il devient impossible de faire confiance entièrement à l'un ou l'autre et un rapprochement des points de vue est alors souhaitable avant la délégation d'une tâche. D'où la question générale que nous abordons dans ce mémoire : peut-on élaborer un modèle « computationnel » de comparaison des points de vue divergents avant la prise de contrôle d'une tâche?

La cohabitation usager-agent diffère de la cohabitation agent-agent par la difficulté à modéliser l'utilisateur et par le fait qu'il n'existe pas de modèle offrant une bijection usager-agent parce que les théories proposées reposent sur des hypothèses réductrices

[Tessier et al., 2001]. Il faut cependant préciser que notre champ d'investigation ne concerne pas la modélisation de l'utilisateur en utilisant les sciences cognitives.

Ce chapitre permettra de présenter l'approche théorique développée lors de cette étude. Cette approche fournira un modèle formel de modélisation et de comparaison des points de vue d'un usager et de son agent. Elle permettra également de tenir compte des problèmes de compréhension usager-agent, liés aux conflits terminologiques. Cette approche s'appuie sur la logique terminologique, décrite en détail au chapitre 3. L'utilisation de cette logique permettra de transformer les problèmes de conflits terminologiques en un problème de classification de concepts dans une taxonomie de concepts.

Dans un premier temps, nous définirons la notion de similarité et nous préciserons le concept de « point de vue ». Ensuite, une présentation formelle du modèle théorique sur lequel s'appuie notre approche sera faite. Finalement, nous ferons une description détaillée du cas concret concernant une application de gestion de stocks en commerce électronique, afin de valider notre approche théorique.

4.2 La notion de similarité

La notion de similarité désigne généralement une relation de ressemblance entre deux ou plusieurs objets [Larousse, 1980]. Elle peut se définir comme la somme des points ou des caractéristiques communes entre deux objets. Mesurer la similarité peut donc se

réduire à déterminer le nombre de caractéristiques communes entre deux objets, comparativement au nombre total de caractéristiques que possèdent ceux-ci [Diday et al., 2000]. Prenons, par exemple, deux états, p et q, impliquant chacun un certain nombre d'autres états :

$$p \Rightarrow (a \wedge b \wedge c \wedge d)$$
$$q \Rightarrow (a \wedge b \wedge e \wedge f)$$

Les états p et q impliquent chacun un certain nombre d'autres états, qui peuvent représenter leurs caractéristiques. De plus, ils impliquent tous les deux les états a et b, ceux-ci constituant leurs caractéristiques communes. On pourrait représenter la similarité entre les états p et q de la façon suivante:

$$\mathrm{SM}_{pq} \equiv (a \wedge b)$$

D'autre part, afin de pouvoir déterminer si deux objets sont similaires, il faut également que ceux-ci possèdent des caractéristiques ou des composantes qui sont comparables. On pourra difficilement affirmer qu'un téléviseur est similaire à une machine à laver, car on ne peut pas vraiment comparer leurs caractéristiques respectives. Dans notre cas, avant de mesurer la similarité des concepts composant les points de vue, nous garantissons leur égalité sémantique en vérifiant l'existence d'une relation de subsomption entre chacun des concepts qui les composent. En définitive, nous définirons donc la similarité comme une mesure quantitative des ressemblances qui existent entre deux objets comparables.

4.3 Concept de point de vue

Un point de vue peut se définir comme étant une représentation quelconque de l'état mental d'une entité (utilisateur ou agent) à propos d'une situation donnée. Il s'agit en fait de l'opinion de l'entité, de sa façon de percevoir un problème et de vouloir le résoudre. Dans le cas présent, nous définirons le point de vue d'une entité à l'aide d'une structure terminologique basée sur le modèle BDI [Rao et Georgeff, 1991]. Nous utiliserons donc les notions de buts, de croyances et de plans d'actions (intentions), telles qu'elles sont définies à la section 2.3 du chapitre 2, pour représenter un point de vue. La figure 4.1 nous montre l'exemple d'un usager et d'un agent, exprimant chacun leur point de vue dans un contexte de gestion des stocks d'une entreprise.

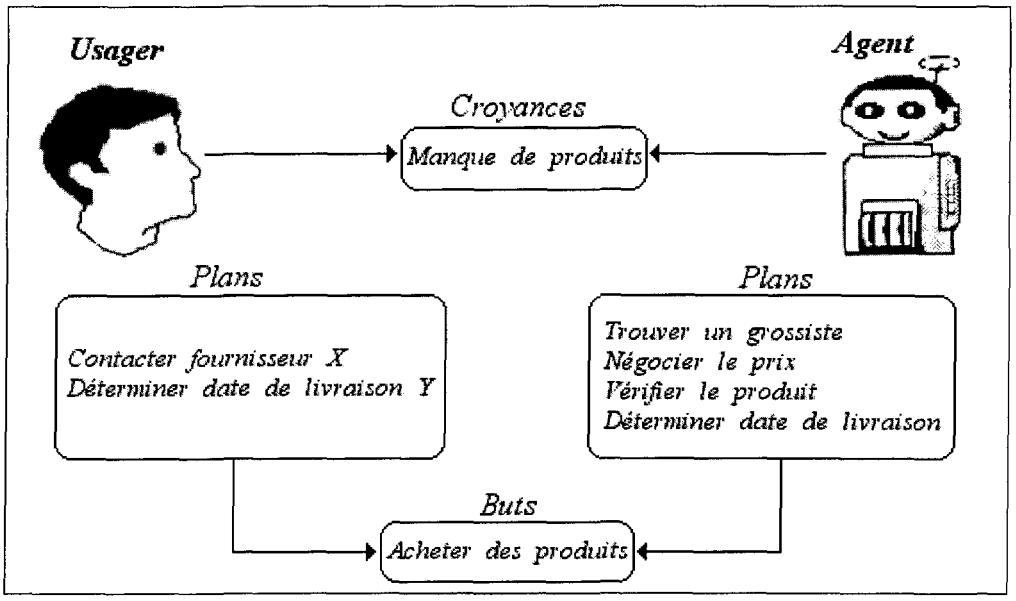

Fig. 4.1 : Concept de point de vue

Dans l'exemple ci-haut, l'utilisateur et l'agent ont exactement la même croyance, c'est-à-dire qu'ils croient tous les deux en un manque de produit, et ont aussi le même objectif, c'est-à-dire de vouloir acheter des produits. Cependant, le plan d'actions de l'agent est seulement partiel, car il a planifié de commander des produits, mais il n'a pas choisi encore de fournisseur. Son plan consiste à trouver un grossiste, à négocier le prix du produit avec celui-ci, de vérifier la qualité du produit et finalement de déterminer une date de livraison. De l'autre côté, l'usager a un plan d'actions moins étoffé, mais complet. Il souhaite contacter un fournisseur « X » qu'il connaît déjà et fixer avec lui une date de livraison « Y ». En d'autres termes, cet exemple nous montre une entité (l'utilisateur) capable de spécifier ses besoins sans être apte à préciser tous les détails de son plan d'actions et son collègue (l'agent), qui lui, peut définir en détail son plan d'actions en prenant ses renseignements dans sa base de données.

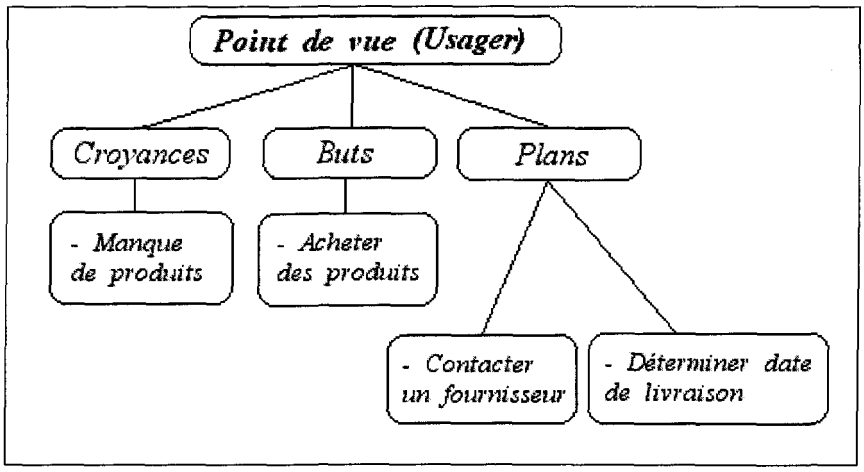

Fig. 4.2 : Structure terminologique de représentation d'un point de vue

Afin de pouvoir manipuler, analyser et comparer les points de vue de l'usager et de l'agent, nous proposons l'utilisation d'une structure terminologique pour la représentation de ceux-ci, tel qu'illustré à la figure 4.2. La représentation d'un point de vue consiste donc en la création en logique terminologique d'un concept racine incluant les différentes parties du point de vue. On peut voir que le point de vue de l'usager se définit comme un concept terminologique incluant trois sous-concepts, nommés « Croyance », « But » et « Plan ». Ces trois concepts sont eux aussi composés de concepts représentant les détails du point de vue de l'entité. En définitive, nous définirons un point de vue comme étant une ontologie représentant une agrégation des concepts définissant les croyances, les buts et les plans d'une entité.

4.4 Approche proposée

L'approche théorique que nous proposons consiste à modéliser les points de vue de l'usager et de son agent en logique terminologique afin de pouvoir classifier chaque concept d'un point de vue donné dans une taxonomie de concepts de l'autre point de vue. La mesure de la similarité entre deux points de vue considèrera donc la correspondance partielle entre les différents concepts qui composent ces points de vue [Bouzouane et Bouchard, 2003]. Notre approche logique utilisera la classification terminologique afin de garantir l'égalité sémantique des points de vue usager-agent, avant l'estimation de la divergence entre ceux-ci. En d'autres termes, on s'assurera d'abord que les deux points de vue sont comparables. La figure 4.3 nous montre la traduction des deux points de vue usager-agent de la figure 4.1 à l'aide de la logique terminologique.

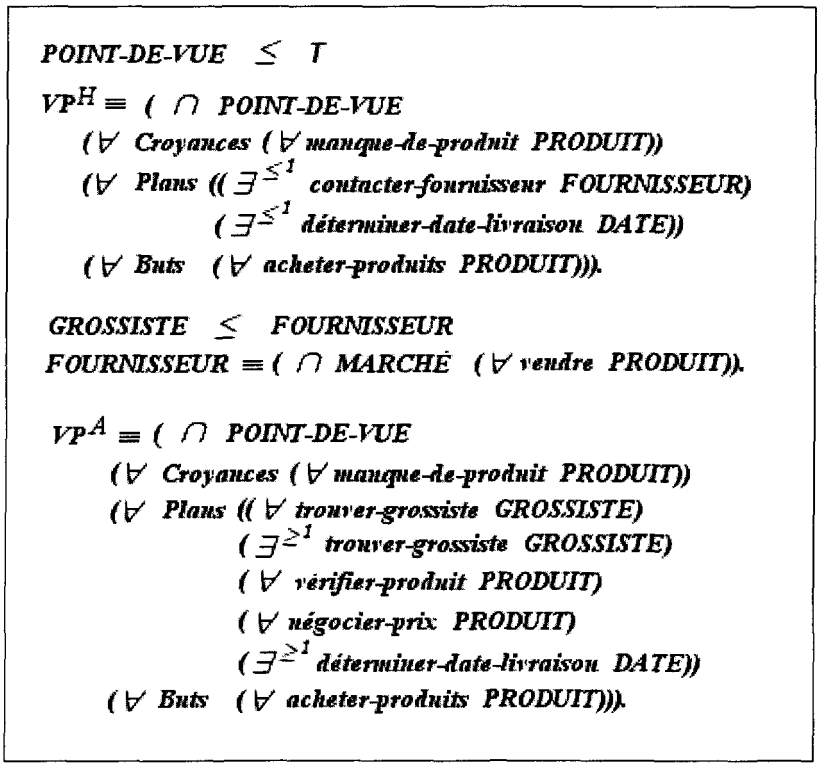

Fig. 4.3 : Traduction des points de vue usager-agent en logique terminologique

Le concept de ***POINT-DE-VUE*** est *subsumé* par tous les concepts de l'univers du discours représenté par le symbole T, qui désigne la racine de la taxonomie ou de l'ontologie. Dans l'exemple ci-haut, le symbole $\forall$ représente l'opérateur *ALL* en Loom et le symbole $\cap$ est équivalent à l'opérateur *AND*. Le point de vue de l'utilisateur, représenté par VP^H, se définit comme étant une conjonction entre un ***POINT-DE-VUE*** et des ***Croyances*** concernant toutes les entités qui entretiennent un rôle ***manque-de-produit***. Les ***Plans*** inclus dans le concept VP^H consistent à contacter au plus un fournisseur spécifique (représenté par la contrainte $\exists^{\leq 1}$, qui est équivalente au constructeur *at-least* en Loom) et à déterminer au plus une date de livraison. L'ensemble des buts de l'usager consiste à acheter

des produits. Le concept ***GROSSISTE*** est défini comme étant la conjonction entre le concept ***MARCHÉ*** et les entités entretenant un rôle ***vendre-produit***. Le concept VP^A est défini exactement de la même façon que VP^H, à l'exception de la partie concernant les plans. Ceux de l'agent consistent à trouver au moins un grossiste (représenté par la contrainte $\exists^{\geq 1}$, qui est équivalente au constructeur *at-most* en Loom), à vérifier les produits vendus par le grossiste, à négocier le prix des produits avec le grossiste trouvé et à déterminer au moins une date de livraison.

Dans l'exemple de la figure 4.3, on peut voir apparaître des conflits terminologiques. Les points de vue des deux entités sont similaires, même s'ils n'utilisent pas toujours les mêmes termes. Par exemple, l'action « contacter-fournisseur » et l'action « trouver-grossiste » partagent une relation de subsomption, telle que spécifiée par la définition ***GROSSISTE*** $\leq$ ***FOURNISSEUR***.

4.5 Formalisation de l'approche

Notre proposition est simple, lorsque l'usager ou l'agent souhaitera prendre une initiative, dans un contexte de planification à initiative mixte, nous proposons de comparer d'abord leurs points de vue respectifs de la situation en utilisant la logique terminologique, pour s'assurer dans un premier temps de l'égalité sémantique de leurs points de vue, et dans un deuxième temps de mesurer la crédibilité de la nouvelle initiative proposée. La mesure de la crédibilité sera basée sur une heuristique de classification qui sera décrite en détails dans les sections suivantes. De cette façon, les conflits terminologiques ainsi que les

initiatives à risque seront évités lors de la délégation du contrôle. Par conséquent, deux concepts similaires, même s'ils ne sont pas identiques, devront être liés par une relation de subsomption. Comme nous l'avons mentionné au chapitre 3, la relation de subsomption permet de classifier un concept parmi une taxonomie d'éléments.

4.5.1 Crédibilité de l'initiative de l'agent

Un usager aura confiance en l'initiative de son agent si son point de vue d'une situation donnée *subsume* celui de l'agent. Cette affirmation se formalise de la façon suivante :

$$\textit{L'usager a confiance en l'initiative de l'agent} \Rightarrow VP^A \leq VP^H \Leftrightarrow$$
$$(\forall C^A \in VP^A, \exists C^H \in VP^H \mid C^A \leq C^H \wedge SFC(C^H, C^A) \in]0,1])$$

C^A représente un concept inféré par l'agent en fonction de son opinion et C^H représente un concept proposé par l'usager. Le facteur de similarité entre les deux concepts, représenté par $SFC(C^A, C^H)$, est évalué lorsqu'une égalité sémantique existe entre ces deux concepts (C^A, C^H). Le calcul de l'égalité sémantique est basé sur la relation de subsomption, notée $C^A \leq C^H$. Le facteur de similarité est utilisé pour déterminer le degré de crédibilité d'une initiative mixte entre l'utilisateur et l'agent artificiel agissant en son nom. Il est défini comme étant une moyenne des facteurs de similarité de chacun des rôles constituants deux concepts. Ce facteur de similarité entre deux concepts n'est pas calculé d'une façon arbitraire ou aléatoire; il est déterminé à l'aide d'une heuristique définie par la formule suivante :

$$SFC(C^H, C^A) = \qquad (1)$$

$$\frac{1}{Max(C^H, C^A)} \sum_{j=1}^{Min(C^H, C^A)} [SFD(rôle_j^H, rôle_j^A)] * [[SFN(rôle_j^H, rôle_j^A) + SFV(rôle_j^H, rôle_j^A)]/2]$$

- ***Min(C^H, C^A)***: représente le nombre minimal de rôles existant entre deux concepts comparés et appartenant à chacun des points de vue.

- ***Max(C^H, C^A)***: représente le nombre maximal de rôles existant entre deux concepts comparés et appartenant à chacun des points de vue.

- ***SFD*** : représente le *Facteur* de *Similarité* du *Domaine* dans lequel il existe une égalité sémantique entre le $j^{ième}$ rôle « $rôle_j^H$ » appartenant à un concept composant le point de vue de l'usager et son homonyme, le $j^{ième}$ rôle « $rôle_j^A$ » faisant partie du point de vue de l'agent. On dénote par D_j^H le domaine du « $rôle_j^H$ » et par D_j^A le domaine du « $rôle_j^A$ ». Ce facteur est égal à $Min(D_j^H, D_j^A) / Max(D_j^H, D_j^A)$, si D_j^H *subsume* D_j^A ou si D_j^A *subsume* D_j^H, et que $j \in [1, Min(D_j^H, D_j^A)]$, sinon SFD sera réduit à *0*.

- ***SFN*** : représente le *Facteur* de *Similarité* entre les *Noms* de deux rôles. Ce facteur sera égal à *1* si les noms des deux rôles sont identiques ou

synonymes. Dans le cas contraire, le facteur SFN sera égal à *0*. À chaque fois qu'une relation de subsomption est détectée entre deux rôles, la base de données des synonymes est mise à jour.

- ***SFV*** **:** représente le *Facteur* de *Similarité* entre les *Valeurs* de deux rôles atomiques. Ce facteur est égal à *1* lorsque les valeurs des deux rôles sont identiques. Si les deux rôles ne sont pas atomiques, l'heuristique (1) présentée ci-haut est utilisée de façon récursive.

Dans le cas présent, avoir confiance en l'initiative de l'agent signifie avoir confiance en son opinion, qui reflète sa capacité à mener à bien une tâche. Le niveau de crédibilité présuppose une croyance mutuelle en l'adoption de la tâche qui est déléguée.

4.5.2 Crédibilité de l'initiative de l'usager

L'agent aura confiance en l'initiative de l'usager si son point de vue *subsume* celui de l'utilisateur. On formalise cette affirmation de la façon suivante :

$$\textit{L'agent a confiance en l'initiative de l'usager} \Rightarrow VP^H \leq VP^A \Leftrightarrow$$

$$(\forall C^H \in VP^H, \exists C^A \in VP^A \mid C^H \leq C^A \wedge SFC(C^H, C^A) \in]0,1])$$

Cette définition est très similaire à celle de la section précédente concernant la crédibilité de l'initiative de l'agent. En fait, la seule différence se situe au niveau du sens de la relation de subsomption entre VP^H et VP^A, qui est inversé.

4.5.3 Crédibilité de l'initiative mixte

Il y a initiative mixte entre l'usager et l'agent si et seulement si leurs points de vue respectifs se *subsument*. On formalise cette affirmation de la manière suivante :

$$\textit{Initiative mixte entre l'usager et son agent} \Rightarrow VP^H \equiv VP^A \Leftrightarrow$$
$$(VP^H \leq VP^A \wedge VP^A \leq VP^H \wedge$$
$$\textit{Crédibilité\_de\_l'initiative\_mixte}(VP^H, VP^A) \in]0,1])$$

La crédibilité d'une initiative est une mesure servant à quantifier la similarité des concepts constituant chacun des points de vue de l'usager et de l'agent. Cette mesure se calcule à l'aide d'une heuristique définie par la formule suivante :

$$\textit{Crédibilité\_de\_l'initiative\_mixte}(VP^H, VP^A) = \frac{1}{Max(VP^H, VP^A)} \sum_{j=1}^{Min(VP^H, VP^A)} SFC(C_j^H, C_j^A) \qquad (2)$$

- La mesure de la crédibilité de l'initiative mixte est représentée par le symbole $\textit{Crédibilité\_de\_l'initiative\_mixte}(VP^H, VP^A)$ et peut prendre des valeurs dans l'intervalle *[0, 1]*.

- $Min(VP^H, VP^A)$ représente le nombre minimum de concepts comparés entre VP^H et VP^A.

- ***Max(VP^H, VP^A)*** représente le nombre maximum de concepts comparés entre VP^H et VP^A.

- Si la crédibilité ***Crédibilité_de_l'initiative_mixte(VP^H, VP^A)*** = *1*, alors les deux points de vue sont équivalents et l'initiative mixte est totale.

- Si la crédibilité ***Crédibilité_de_l'initiative_mixte(VP^H, VP^A)*** < *1*, alors les deux points de vue sont divergents et l'initiative mixte est partielle.

La crédibilité d'une initiative mixte n'est pas une mesure subjective. Au contraire, elle est basée sur les attitudes mentales respectives de l'utilisateur et de l'agent, dans le but de supporter la décision de la commutation du contrôle de la tâche commune.

4.5.4 Le risque d'une initiative

La prise d'une initiative par l'une des entités lors de la réalisation d'une tâche conjointe comporte un niveau de risque « ρ » associé à cette tâche. Si la contrainte suivante est respectée: ***Crédibilité_de_l'initiative_mixte(VP^H, VP^A)*** < ρ, alors l'initiative sera refusée. Dans ce cas, le système devra réussir à intégrer les points de vue de l'usager et de son agent jusqu'à ce que la mesure de crédibilité puisse satisfaire la contrainte: ***Crédibilité_de_l'initiative_mixte(VP^H, VP^A)*** $\geq \rho$. En nous référant à notre exemple en e-commerce de la figure 4.3, si l'usager souhaite prendre une initiative en suggérant son plan d'actions et que la tâche soit sous le contrôle de l'agent, cette initiative lui sera refusée, car

son point de vue *subsume* celui de l'agent. À l'inverse, dans le même exemple, si l'agent souhaite prendre l'initiative et que la tâche soit sous le contrôle de l'utilisateur, celle-ci lui sera accordée, car son point de vue est *subsumé* par celui de l'usager. La mesure de la crédibilité permet donc de déterminer si une initiative est sécuritaire pour la prise de contrôle d'une tâche commune ou la délégation de celle-ci.

4.6 Validation de l'approche proposée

Le cas concret servant de validation à notre approche théorique concerne la gestion des stocks dans un contexte de commerce électronique. Durant les dernières années, plusieurs travaux ont été menés afin de créer un système de marché électronique, permettant l'achat et la vente de produits sur le Web sans l'intervention d'opérateurs humains. L'idée était de permettre à des agents intelligents de transiger entre eux. Par exemple, ces agents pourraient être les représentants de diverses entreprises agissant en leurs noms. Ces agents se déplacent sur le réseau, s'inscrivent sur le marché et transigent des produits. Un exemple de réalisation dans ce domaine est le système de marché électronique Kasbah [Chavez et Pattie, 1996], qui a servi de source d'inspiration à notre application.

4.6.1 Architecture de l'application

On peut voir à la figure 4.4 l'architecture complète de notre système de marché électronique. Les deux carrés à gauche et à droite de l'image correspondent respectivement aux parties client et serveur qui composent le système.

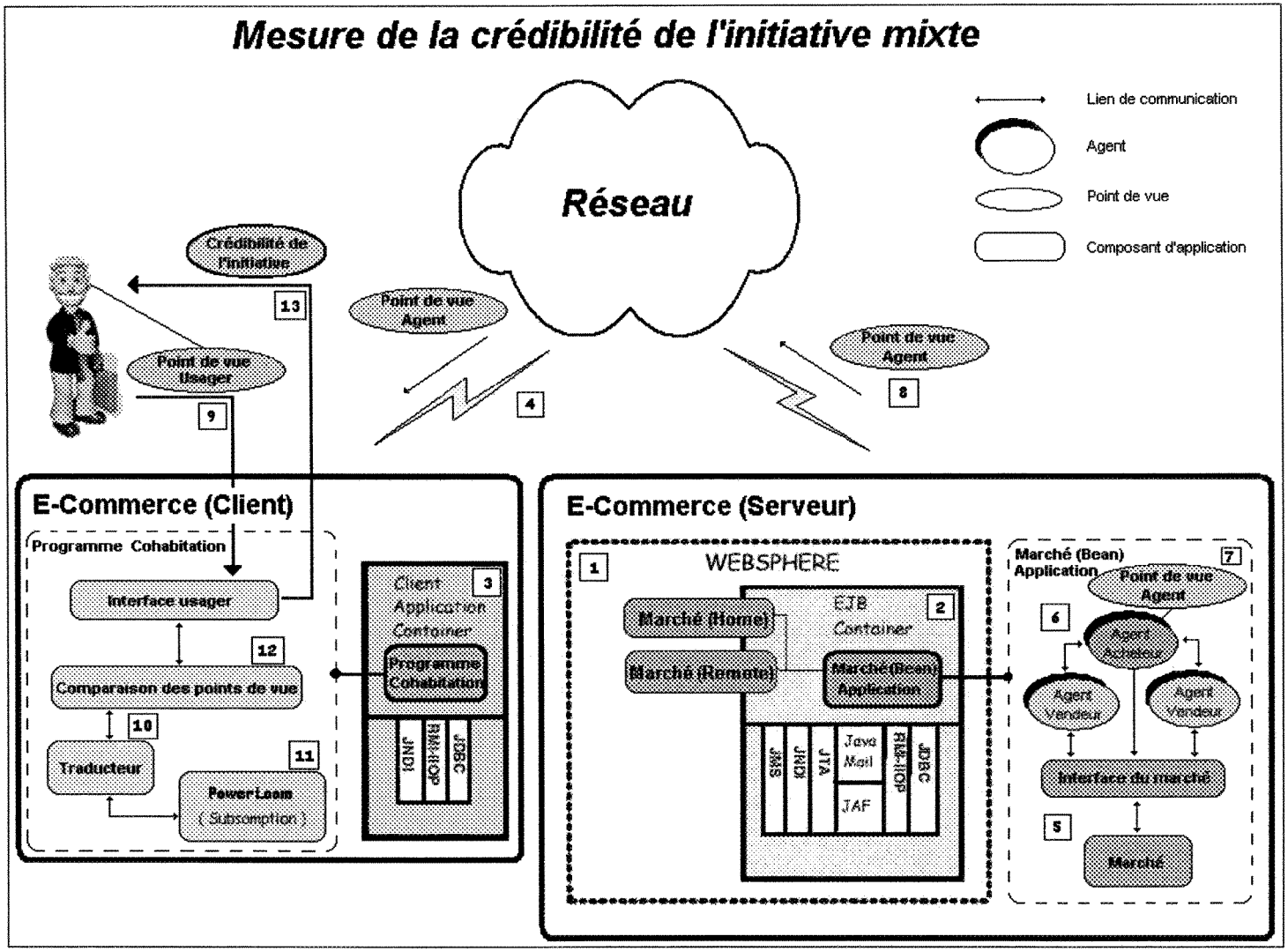

Fig. 4.4 : Architecture du système de marché électronique

4.6.1.1 Le serveur de e-commerce

La partie serveur constitue le marché électronique dans son ensemble. Elle inclut les agents, l'interface de communication entre les agents ainsi que le marché lui-même. Cette partie est construite à l'intérieur d'un composant EJB (Entreprise Java Bean) [Deitel et Deitel, 2000] qui s'exécute via un serveur de e-commerce IBM WebSphere [WebSphere, 2003]. Ce serveur de e-commerce gère toutes les communications entre la partie client et les différents composants ainsi que la sécurité. Pour ce faire, il utilise la technologie RMI [Deitel et Deitel, 2000]. Les agents présents sur le serveur sont créés directement sur celui-ci, ils ne se déplacent pas. C'est l'application client qui doit faire une demande à la partie serveur afin que celle-ci crée un agent sur le marché. Un certain nombre de paramètres, tels le type de l'agent (vendeur ou acheteur), le type de produit à transiger, les différents barèmes de prix ainsi que les renseignements concernant le produit, devront être envoyés au marché afin que celui-ci crée un agent répondant aux critères désirés. Le marché renverra ensuite un numéro d'identification de l'agent à l'application client, lui permettant ainsi de contacter directement le nouvel agent.

Lorsque le serveur de marché électronique est démarré, il attend simplement les connexions et les requêtes des applications clients et gère les agents qui sont présents sur celui-ci. Il a également pour mission de mettre à jour les informations sur les agents et de répondre aux différentes requêtes via son interface. La liaison entre les applications clients et le marché est prise en charge par le serveur WebSphere.

Lorsqu'un utilisateur souhaitera introduire un agent sur le marché, il devra d'abord démarrer une instance de l'application client. Celle-ci se connectera automatiquement au serveur de e-commerce. La connexion s'effectuera grâce aux trois composants du marché électronique que l'on peut voir à la figure 4.4, soient les composants *Home*, *Remote* et *Bean*.

- **Composant Home**: ce composant statique sert de balise, de point de repère pour l'application client. Lorsqu'une application client en Java souhaite communiquer avec un composant EJB, elle doit d'abord se connecter à une adresse réseau (par exemple une adresse IP) via les méthodes du package RMI. Par la suite, elle obtiendra un pointeur référence lui permettant d'invoquer les méthodes statiques du composant *Home*, qui est chargé de la construction effective des autres composants, de leurs destructions et de leurs recherches. L'application client devra invoquer le composant *Home*, afin que celui-ci crée un composant *Remote*. Ce composant lui permettra par la suite de communiquer directement avec le marché électronique et d'effectuer des requêtes.

- **Composant Remote** : ce composant constitue l'interface utilisable par l'application client. Cette classe sera seulement constituée des entêtes des méthodes disponibles à distance. Elle ne contiendra pas l'implémentation réelle du code de programmation. Ce composant servira de « commande à distance » pour la communication client-

serveur une fois la connexion effectuée. Il peut exister plusieurs composants *Remote*, qui sont chacun en lien avec une et une seule application client.

- **Composant Bean :** ce composant inclut toutes les méthodes et les fonctions de l'application serveur, qu'elles soient disponibles à distance ou non, et l'implémentation du code de programmation qui s'y rattache. C'est en fait le cœur du composant EJB. Dans le cas présent, il s'agit du marché électronique. Lorsque l'application client invoque une méthode via un composant *Remote*, celui-ci la fait exécuter par le composant *Bean* et retourne la réponse au client. Il n'y a qu'une seule instance du composant *Bean* en action, tous les composants *Remote* se référant au même composant *Bean.*

Une fois la connexion client-serveur effectuée, l'application client peut utiliser les méthodes du serveur de la même façon que si elles étaient situées localement. La répartition client-serveur devient donc transparente pour l'utilisateur (opérateur humain) et pour l'application client. L'usager pourra alors faire une requête pour la création d'un agent selon les paramètres qu'il désire. Il disposera ensuite d'un agent qui agira comme son représentant sur le marché. Il est à noter que plusieurs utilisateurs peuvent démarrer simultanément plusieurs applications clients sur différents ordinateurs et ainsi peupler le marché d'une multitude d'agents qui échangeront entre eux. Le fonctionnement des différents types d'agents sera décrit dans la section 4.6.4 du chapitre.

4.6.1.2 L'application client

La partie client constitue l'outil de l'utilisateur. Celui-ci, grâce à l'interface de l'application client, se connecte au marché, demande la création d'un agent ou visualise ce qui se passe sur le marché. L'interface de l'application client, que l'on peut voir à la figure 4.5, permet également à l'utilisateur de contacter son agent, afin d'obtenir son point de vue sur la situation du marché. Il pourra ensuite comparer celui-ci avec son propre point de vue.

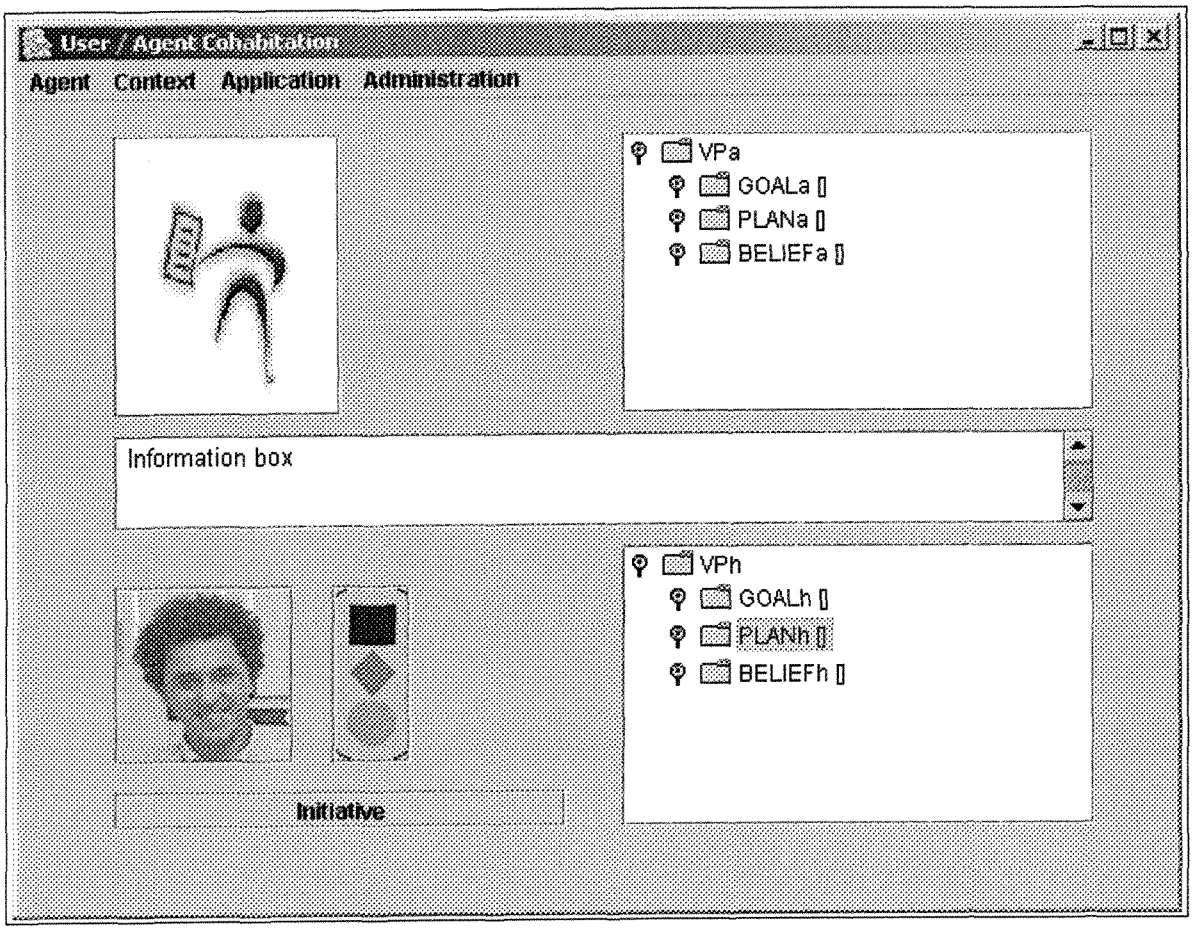

Fig. 4.5 : Interface de l'application client

Nous avons conçu l'application client entièrement en Java. Elle est composée d'un module de comparaison des points de vue, basé sur la logique terminologique et sur le modèle théorique présenté à la section 4.5. Ce module inclut un traducteur de point de vue permettant de traduire un formalisme objet en logique terminologique, comme nous le montre la figure 4.4, ainsi qu'un moteur d'inférence PowerLoom [PowerLoom 1997] pour la classification. Les informations concernant l'agent sont présentées dans la partie supérieure de l'écran et celles de l'utilisateur sont dans la partie inférieure. L'utilisateur peut connaître en temps réel le point de vue de son agent sur la situation du marché ainsi que son plan d'actions, à l'aide du panneau en haut à droite. Il peut également voir l'activité sur le marché grâce au menu « Administration », comme nous le montre la figure 4.6.

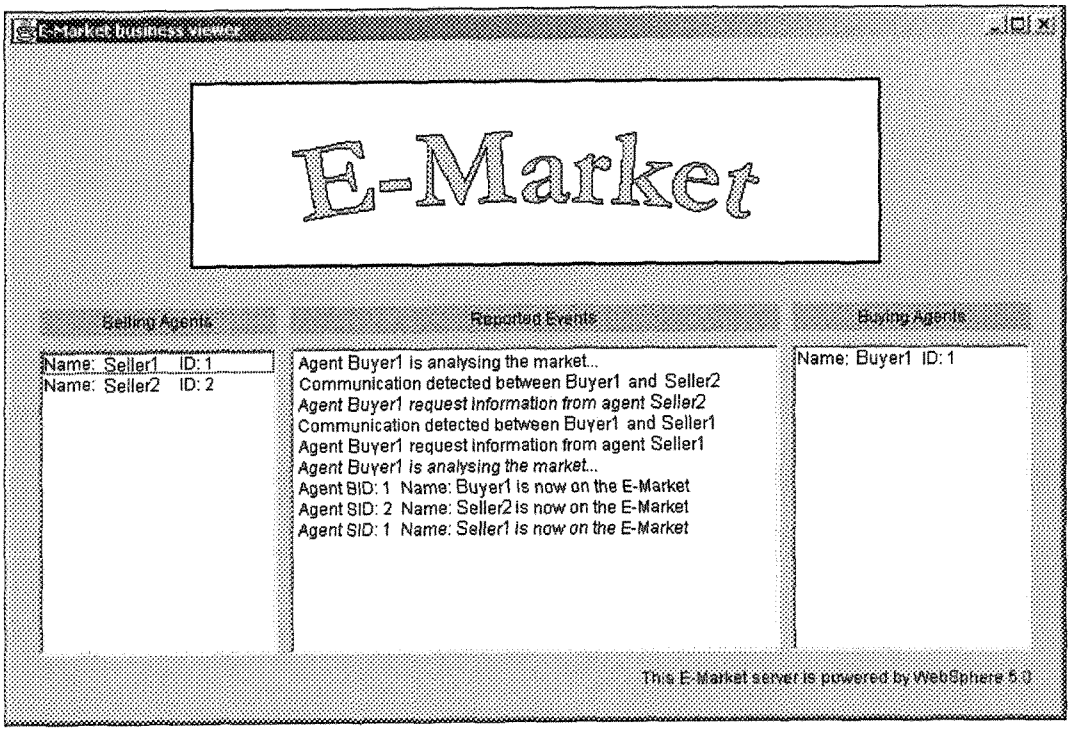

Fig. 4.6: Visualisation de l'activité sur le marché électronique

À tout moment, durant le déroulement des activités, l'utilisateur peut tenter de prendre une initiative et de soumettre son plan d'actions grâce au bouton « Initiative ». Pour cela, il devra d'abord modéliser son point de vue dans la boîte, en bas à droite, à l'aide d'un outil de création en Java, que l'on peut voir à la figure 4.7. Ce point de vue sera ensuite transformé par le composant traducteur en logique terminologique, en utilisant le formalisme de PowerLoom.

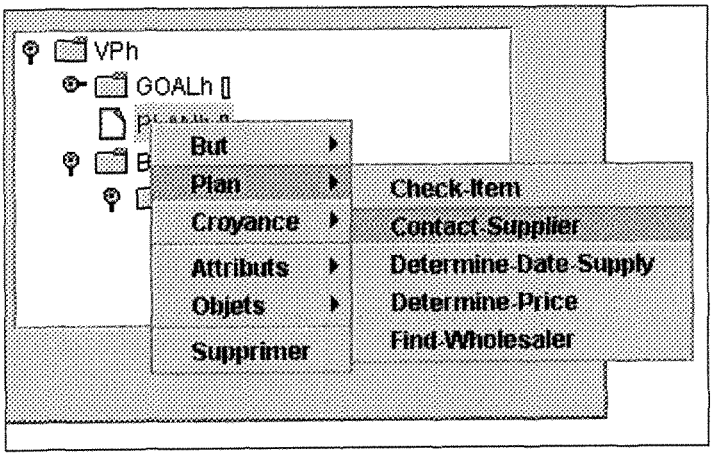

Fig. 4.7 : Outil de création de points de vue de l'usager

Avant que le système permette l'initiative, il s'assurera de l'égalité sémantique entre les deux points de vue. Pour cela, il questionnera simplement le moteur d'inférence de PowerLoom pour vérifier s'il existe une relation de subsomption entre le concept « VPa » et le concept « VPh ». Ensuite, il effectuera la comparaison des points de vue, suivant les heuristiques présentées à la section 4.5. Cette comparaison a pour but de valider la crédibilité de l'initiative proposée. Si l'initiative est acceptée, le panneau de signalisation tournera au vert, dans le cas contraire, il tournera au rouge. Durant la conception des points

de vue, le panneau sera au jaune. Comme nous l'avons vu à la section 4.5, l'initiative sera acceptée uniquement si la mesure de la crédibilité est supérieure au niveau de risque ρ. Il est donc possible de définir le niveau de risque ρ de la tâche en cours à l'aide du menu « context » disponible en haut de l'écran, tel qu'on peut le voir sur la figure 4.8.

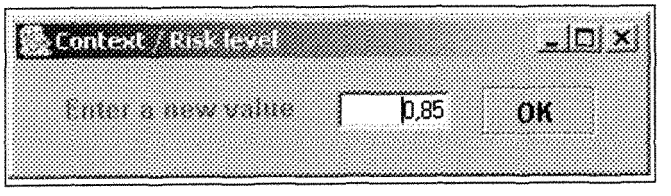

Fig. 4.8 : Panneau de modification du niveau de niveau de risque ρ

4.6.2 PowerLoom

PowerLoom est le successeur du système de représentation de la connaissance Loom [Loom, 1991], disponible en version Java ou C++. Il fournit un langage et un environnement permettant de construire des applications intelligentes de classification, dotées de base de connaissances terminologique [PowerLoom, 1997]. PowerLoom emploie un formalisme dérivé de celui de Loom, tel qu'on peut le voir sur la figure 4.9, qui est similaire et de même puissance d'expression. Son moteur d'inférence utilise un modèle de déduction par chaînage avant [Farreny et Ghallab, 1987] qui peut manipuler des règles et des relations complexes, telles les relations de subsomption. PowerLoom est également munit d'un système de classification avancé, permettant de positionner avec beaucoup de précision un nouveau concept dans une hiérarchie déjà existante. Il s'intègre facilement à une autre application, comme un composant permettant la création de taxonomie de

concepts. Cette intégration s'effectue grâce à une API (Application Programming Interface) permettant de créer et de manipuler la base de connaissances PowerLoom via une autre application. PowerLoom devient ensuite un package faisant partie intégrante de l'application.

LOOM	*PowerLoom*
(defconcept ETRE :is :primitive) *(defconcept ENSEMBLE :is :primitive)* *(defconcept HUMAIN :is (:and ETRE :primitive))* *(defconcept HOMME :is (:and HUMAIN :primitive))*	*(defconcept ETRE (?x))* *(defconcept ENSEMBLE (?x))* *(defconcept HUMAIN (?x ETRE))* *(defconcept HOMME (?x HUMAIN))*
(defconcept FEMME :is (:and HUMAIN *(:not HOMME) :primitive))*	*(defconcept FEMME (?x HUMAIN):<=>* *(and (not (HOMME ?x))))*
(defconcept ROBOT :is (:and ETRE *(:not HUMAIN) :primitive))*	*(defconcept ROBOT (?x ETRE):<=>* *(and (not (HUMAIN ?x))))*
(defrelation membre :is :primitive)) *(defrelation chef :is (:and membre :primitive))*	*(defrelation membre ((?x) (?y)))* *(defrelation chef ((?x) (?y)):<=>* *(and (membre ?x ?y)))*
(defconcept EQUIPE :is (:and ENSEMBLE *(:all membre ETRE)* *(:at-least 2 membre)))*	*(defconcept EQUIPE (?x):<=>* *(and (ENSEMBLE ?x)* *(all (membre ?x ?y) (ETRE ?x))* *(at-least 2 (membre ?x ?y))))*
(defconcept PETITE-EQUIPE :is (:and EQUIPE *(:at-most 5 membre)))*	*(defconcept PETITE-EQUIPE (?x):<=>* *(and (EQUIPE ?x)* *(at-most 5 (membre ?x ?y))))*
(defconcept EQUIPE-MODERNE :is (:and EQUIPE *(:at-most 4 membre)* *(:at-least 1 chef)* *(:all chef ROBOT))*	*(defconcept EQUIPE-MODERNE (?x):<=>* *(and (EQUIPE ?x)* *(at-most 4 (membre ?x ?y))* *(at-least 1 (chef ?x ?w))* *(ROBOT ?w)))*

Fig. 4.9 : Comparaison des formalismes LOOM et PowerLoom

On peut noter sur la figure 4.9 les différences entre le formalisme original de LOOM et celui de son successeur PowerLoom. On peut voir que l'introduction de la

définition du concept se fait grâce au symbole «<=>», contrairement au constructeur « is » de LOOM. D'autre part, on note qu'à chaque introduction d'un concept ou d'un rôle, on doit inclure des variables représentant les instances possibles de ceux-ci. Ces variables servent de références lors de la définition des concepts et sont désignées par le symbole « ? ». Outre ces quelques différences, on peut voir que les deux formalismes sont très similaires.

4.6.3 Conceptualisation des points de vue en logique terminologique

Dans l'application client, les points de vue de l'agent et de l'utilisateur sont d'abord créés à l'aide d'objets en Java. Ces points de vue sont créés avec une structure d'arbre permettant à l'utilisateur et à son agent de définir leurs croyances, leurs plans d'actions et leurs buts. Prenons, par exemple, le point de vue de l'usager tel que décrit à la figure 4.10. La racine de l'arbre est nommée « VPh », elle représente le concept du point de vue de l'usager. Un point de vue doit obligatoirement contenir trois sous-concepts, représentant les buts, les croyances et les plans de l'entité, nommé respectivement « GOALh », « BELIEFh » et « PLANh ». Chacun de ces concepts représente une section qui contient les éléments qui ont été introduits par l'utilisateur. Un élément est toujours constitué d'une relation et d'un concept qui lui est rattaché. Par exemple, le premier et unique élément de la section « BELIEFh » est constitué de la relation « Missing-Product », qui implique un concept « Software », et qui désigne la croyance de l'utilisateur concernant le manque de logiciel. En clair, l'exemple de la figure 4.10 signifie que l'utilisateur croit qu'il manque

actuellement de logiciels en stock, il a donc pour objectif de s'en procurer, et son plan d'actions pour y arriver consiste à contacter un fournisseur nommé « FutureShop ».

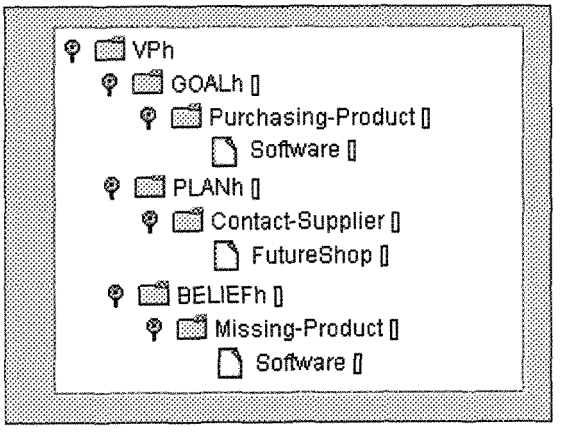

Fig. 4.10 : Exemple d'un point de vue modélisé en Java

Une fois que les points de vue de l'usager et de l'agent ont été modélisés avec des objets Java, ceux-ci devront être traduits en logique terminologique, en utilisant le formalisme de PowerLoom. Pour cela, la représentation structurelle des points de vue sera convertie en concepts terminologiques. Les différents éléments contenus dans chacune des sections seront transformés en deux éléments chacun, soit un concept et un rôle. Par exemple, l'élément de la section « GOALh », à la figure 4.10, sera converti en un concept nommé « Software » et un rôle nommé « Purchasing-Product », qui désignera une relation entre le concept « PLANh » et le concept « Software », tel qu'illustré à la figure 4.11.

Structure en JAVA	*Définition terminologique*
GOALh [] Purchasing-Product [] Software []	*(defconcept Goalh (?x))* *(defconcept Software (?y))* *(defrelation purchasing-product* *((?x Goalh) (?y Software)):<=> (Software ?y))*

Fig. 4.11 : Traduction d'une partie d'un point de vue en logique terminologique

4.6.3.1 Le traducteur de points de vue

Le traducteur de points de vue, inclus dans l'application client, a pour rôle de traduire un point de vue complet, représenté sous forme d'une arborescence en Java, en une série de définitions terminologiques équivalentes. Un exemple de fichier de traduction généré par l'application client est disponible à l'annexe 1. L'exemple de la figure 4.12 nous montre la traduction complète du point de vue de l'utilisateur de la figure 4.10, en une série de définitions terminologiques, utilisant le formalisme de PowerLoom. Ce traducteur est relié à PowerLoom via une API, permettant ainsi le dialogue direct entre celui-ci et la base de connaissances. Chaque fois qu'une définition de concept est envoyée à PowerLoom, celui-ci classifie ce nouveau concept parmi la taxonomie de concepts déjà existants.

```
(defconcept VPH (?x):<=> (exists (?p ?b ?g) (and (Beliefh ?b) (Planh ?p) (Goalh ?g))))

(defconcept Planh (?x))
(defconcept Beliefh (?x))
(defconcept Goalh (?x))

(defrelation purchasing-product ((?x Goalh) (?y Product)):<=> (Product ?y))
(defrelation contact-supplier ((?x Planh) (?y Supplier)):<=> (Supplier ?y))
(defrelation missing-product ((?x Beliefh) (?y Product)):<=> (Product ?y))
```

Fig. 4.12 : Traduction d'un point de vue avec le formalisme de PowerLoom

Lorsque toutes les définitions concernant les deux points de vue ont été transmises à PowerLoom, l'application client peut alors interroger la base de connaissances en utilisant son algorithme de détection des relations de subsomption, que nous avons détaillé à la section 3.3.2.2 du chapitre 3, afin de déduire si le concept « VPa », qui représente le point de vue de l'agent, *subsume* le concept « VPh », qui représente le point de vue de l'usager et vice-versa. Ce test permet de vérifier l'égalité sémantique des deux points de vue. Une fois l'égalité sémantique assurée, le calcul de la crédibilité de l'initiative pourra alors être effectué en utilisant le modèle théorique présenté à la section 4.5. Il est à noter qu'il est possible de définir préalablement certaines relations de subsomption à l'aide du menu « Administration ». Ces relations de subsomption imposées à l'avance peuvent servir à préciser certains concepts ou rôles, comme nous le montre la figure 4.13.

Chaque fois qu'une relation de subsomption est spécifiée via du menu de la figure 4.13, une requête est envoyée à PowerLoom afin que celui-ci introduise cette définition dans la base de connaissances. Par exemple, la relation de subsomption, définie à la figure 4.13, produirait la requête suivante : *(assert (subset-of Supplier Wholesaler))*. Il s'agit simplement d'une requête définissant l'assertion d'un nouveau fait dans la base de connaissances, spécifiant la relation de subsomption entre le concept « Supplier » et le concept « Wholesaler ».

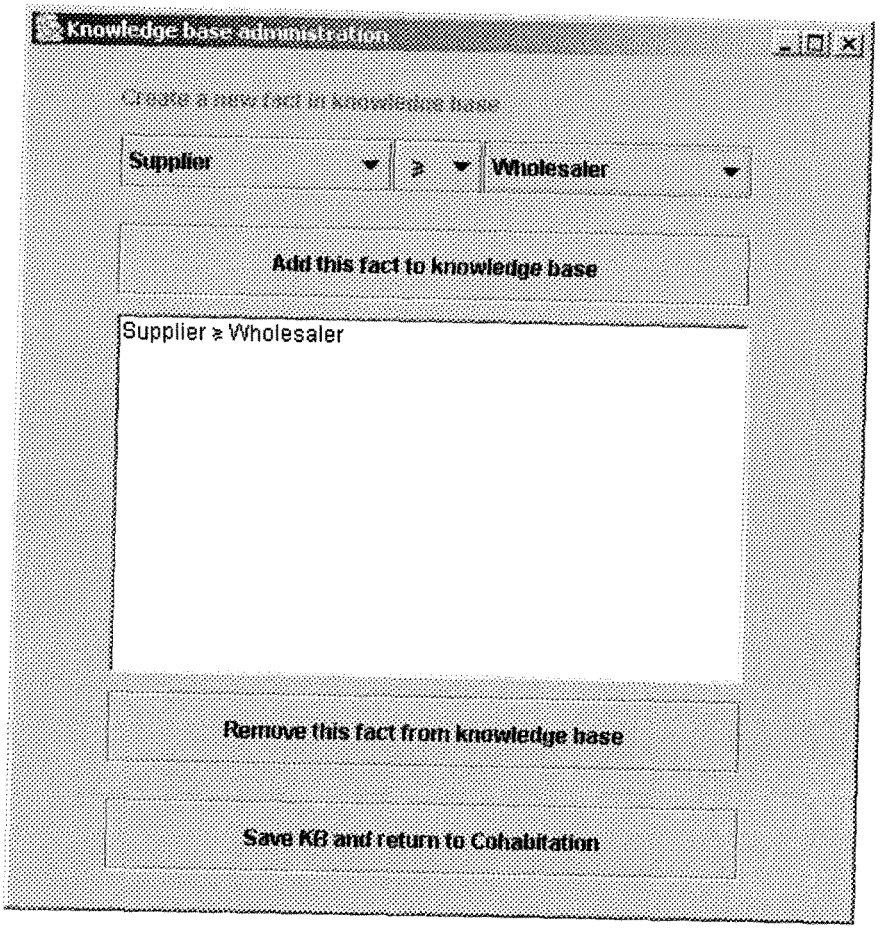

Fig. 4.13 : Définition préalable d'une relation de subsomption

4.6.4 Notion d'agent acheteur et d'agent vendeur

Le serveur de commerce électronique supporte deux types d'agents bien distincts : les agents acheteurs et les agents vendeurs. Ces deux types d'agents peuvent s'inscrire sur le marché, interroger les autres agents et transiger avec eux. La différence majeure entre les agents acheteurs et les agents vendeurs réside dans leurs objectifs primaires : l'acheteur a comme but de se procurer un certain produit et le vendeur a pour but de vendre un certain produit. Chaque agent possède son point de vue de la situation actuelle, qui évolue en

fonction des actions qui se réalisent sur le marché. À tout moment, le créateur de l'agent peut demander le point de vue de celui-ci, tel qu'on peut le voir sur la figure 4.15, s'inscrivant ainsi dans le cycle d'exécution de l'agent.

- **Agent acheteur :** ce type d'agent se verra assigner un certain produit à acheter, avec certaines contraintes comme le prix d'achat maximal et la date limite pour son obtention. Il tentera ensuite d'identifier quels sont les fournisseurs potentiels de ce produit et lesquels offrent les produits et les prix les plus avantageux. Il essayera ensuite de les contacter et de conclure la meilleure transaction possible avec l'un d'entre eux. Son comportement est réglementé grâce à des règles de production, constituant sa base de connaissances, tel qu'on peut le voir à la figure 4.14. On peut voir sur cette figure que l'agent acheteur ne contactera un agent vendeur que si celui-ci vend un produit de bonne qualité, et que la marque du produit en question est connue ou bien que son prix soit négociable.

```
IF produit(i).marque = "connue" AND produit(i).qualite = "bonne"
    ⇒ ContacterVendeur(produit(i))

IF produit(i).marque = "inconnue" AND produit(i).qualite = "bonne" AND
    IF produit(i).negociable = TRUE
        ⇒ ContacterVendeur(produit(i)) AND
          NegocierPrix(produit(i))
```

Fig. 4.14 : Exemple de quelques règles de production d'un agent acheteur

- **Agent vendeur :** ce type d'agent se verra assigner un certain produit à vendre, avec certaines caractéristiques comme la qualité du produit, le prix de vente minimal, la date limite de liquidation, etc. Il tentera ensuite d'identifier quels sont les acheteurs potentiels pour son produit, de les contacter et de conclure la meilleure transaction possible avec l'un d'entre eux. Le comportement de l'agent est formulé sous forme de règles de production.

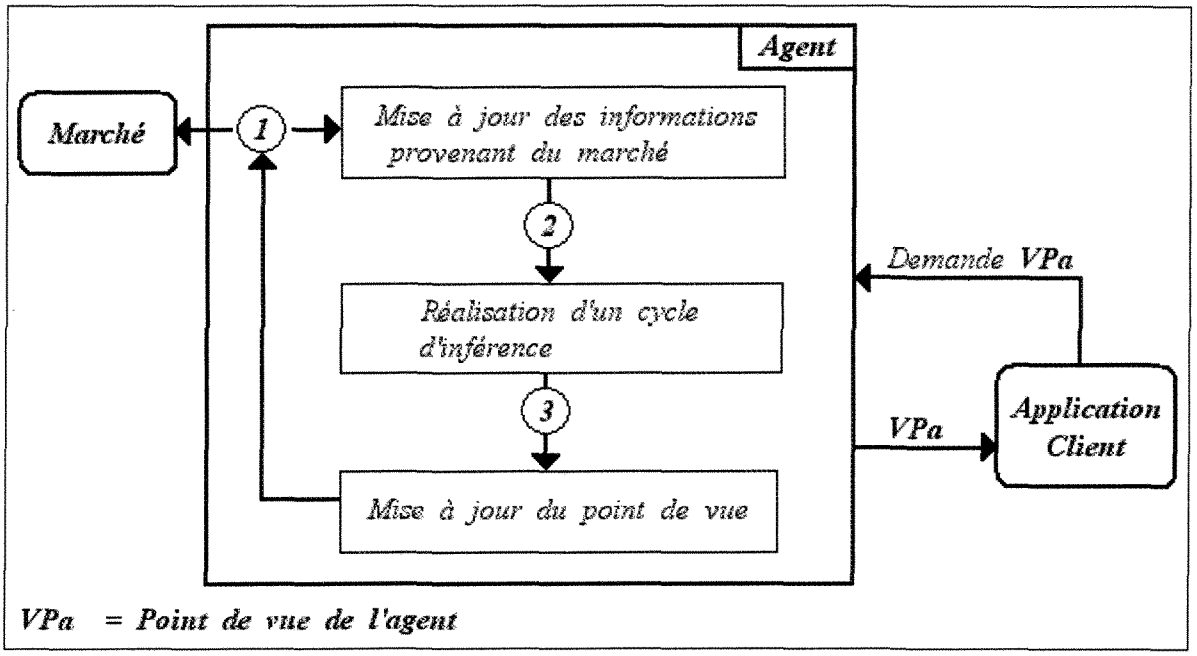

Fig. 4.15 : Cycle d'exécution de l'agent

La figure 4.15 nous montre le cycle d'exécution d'un agent. Une fois inscrit et connecté sur le marché, un agent vendeur ou acheteur met d'abord à jour les informations provenant du marché, comme nous le montre l'étape (1) de la figure 4.15. Cette opération

consiste en une simple requête que l'agent transmet au marché. Celui-ci lui retourne ensuite une structure de données contenant toutes les informations disponibles concernant les agents et l'état du marché. L'agent effectue ensuite un cycle d'inférence, en utilisant ses règles de production, permettant ainsi de déduire de nouvelles connaissances ainsi que son plan d'actions à partir des informations connues du marché (étape 2). Lors de la réalisation d'un cycle d'inférence, les agents peuvent exécuter à tout moment un certain nombre d'actions. La figure 4.16 nous montre la liste de ces actions. Finalement, l'agent mettra à jour son point de vue personnel (étape 3) sur la situation, en construisant une structure de données sous forme d'arborescence, avec des objets Java, à partir des connaissances qu'il a préalablement inférées. Cette structure sera conforme à celle de la figure 4.10.

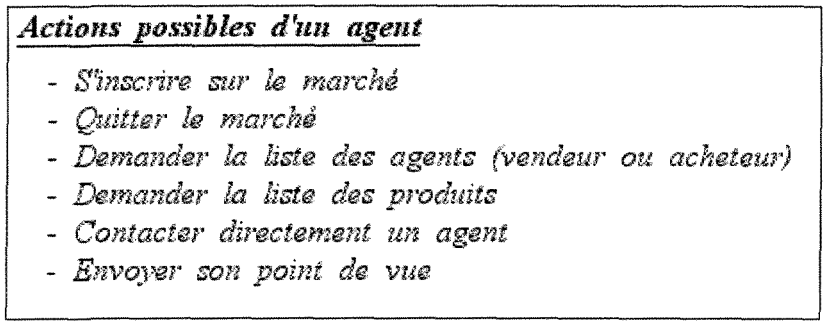

Fig. 4.16 : Liste des actions possibles pour un agent

Lorsqu'il a achevé ses activités, un agent peut décider de quitter le marché, lors de son cycle d'inférence. Il peut également, durant un cycle, consulter la liste des agents potentiels, contacter directement un agent afin de conclure une entente avec celui-ci, et

finalement, il peut envoyer son point de vue à son créateur lorsqu'il en a reçu préalablement la requête.

4.7 Comparaison des points de vue usager-agent

La figure 4.17 nous montre un exemple de résultat obtenu lors de la comparaison des points de vue d'un utilisateur et d'un agent avec notre application. Dans cet exemple, les croyances et les buts des deux entités sont les mêmes. Ils croient tous les deux qu'ils manquent de produits en stock et ils ont comme objectif de s'en procurer. Cependant, le plan d'actions inféré par l'agent diverge de celui proposé par l'utilisateur. L'usager prévoit contacter un fournisseur spécifique, tel qu'indiqué dans son plan d'actions avec « Contact-Supplier ». L'agent prévoit de son côté de trouver un grossiste « Find-Wholesaler », de vérifier la qualité du produit « Check-Item » et de déterminer le prix d'achat avec celui-ci « Determine-Price ». Les deux entités ont ensuite la même idée, soit de déterminer une date de livraison avec l'action « Determine-Date-Supply ».

La vérification de la relation de subsomption entre « VPh » et « VPa » démontre qu'il y a une égalité sémantique entre les deux points de vue, permettant ainsi de les comparer et de mesurer la crédibilité de l'initiative mixte. En effet, le point de vue de l'usager *subsume* celui de l'agent, car chacune des parties de celui-ci possède son équivalent sémantique dans le point de vue de l'agent. L'action « Contact-Supplier » *subsume* l'action « Find-Wholesaler », car le concept « Supplier » *subsume* le concept « Wholesaler », tel que spécifié par une relation de subsomption prédéfinie.

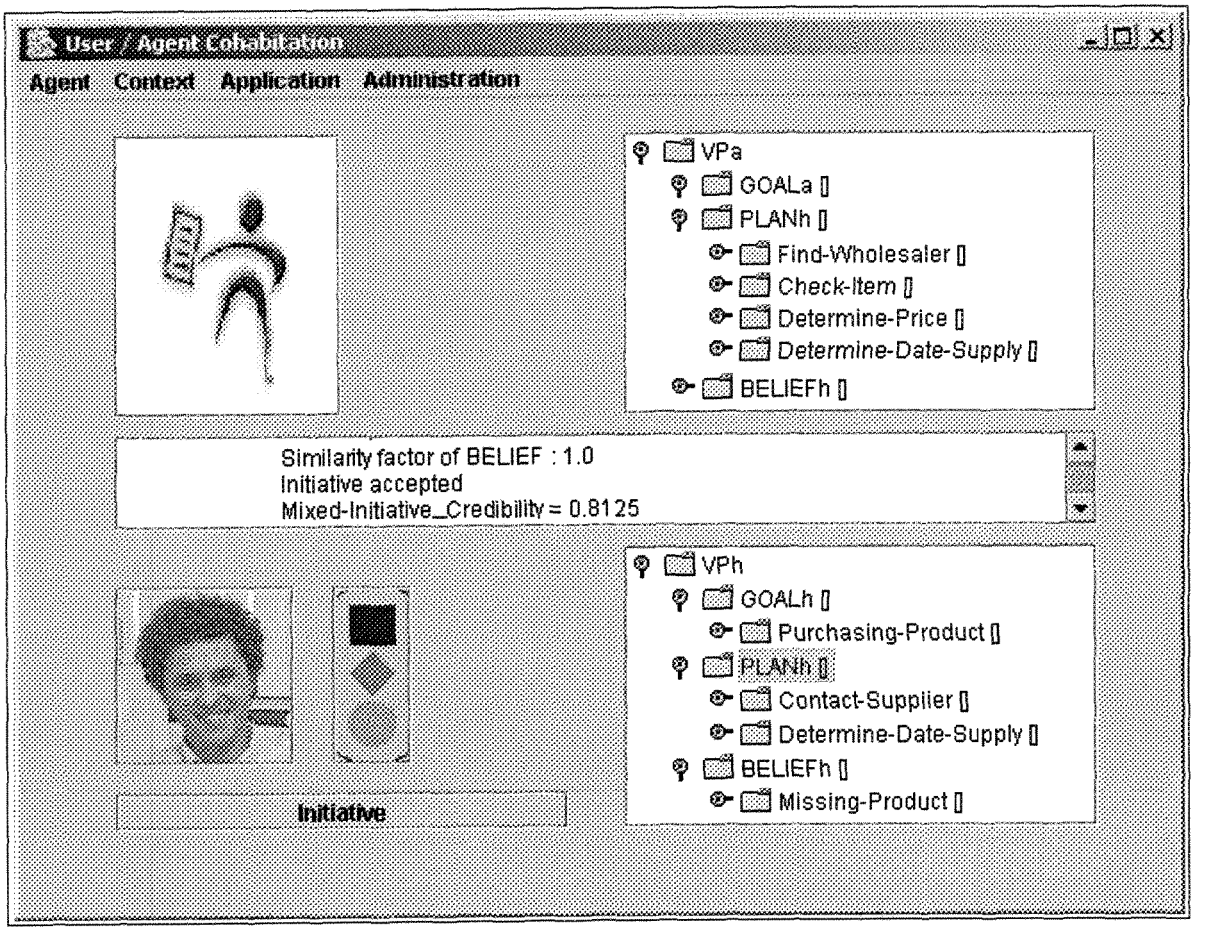

Fig. 4.17 : Exemple de comparaison des points de vue usager-agent

Conformément à l'heuristique (2) présentée à la section 4.5 du chapitre, la crédibilité de l'initiative mixte dans le cas présent est de 81%. Les deux points de vue sont donc divergents à 19% ***(1 – Crédibilité_de_l'initiative_mixte(VP^H, VP^A))***, mais ils convergent à 81%. Cette divergence s'explique par les deux actions supplémentaires proposées par l'agent, soit la vérification de la qualité du produit « Check-Item » et la négociation du prix « Determine-Price ». Du point de vue de l'utilisateur, ces deux actions ne sont pas significatives ou sont implicites. Les croyances et les buts inclus dans les deux

points de vue sont similaires à 100%. Le résultat du calcul nous permet d'affirmer qu'il y a une possibilité de cohabitation entre l'usager et son agent, car la valeur de crédibilité de l'initiative mixte (81%) est substantiellement plus élevée que la valeur de leurs divergences (19%). On peut d'ailleurs voir dans l'exemple de la figure 4.17 que l'initiative a été acceptée, car le niveau de risque ρ relié à la tâche était fixé à 60%, ce qui est inférieur à la valeur de la crédibilité de l'initiative. Il faut également noter que les actions « Find-Wholesaler » et « Contact-Supplier » n'ont pas été considérées, lors du calcul, comme des actions divergentes, à cause de leur relation de subsomption.

4.8 Conclusion

Ce chapitre a permis de présenter un nouveau modèle théorique de comparaison entre les points de vue d'un usager et d'un agent, basé sur la logique terminologique. L'approche présentée dans ce chapitre fournit un modèle « computationel » permettant de calculer d'une manière objective la similarité entre deux points de vue et d'éviter les problèmes de conflits terminologiques lors des interactions usager-agent. Il permet également d'effectuer la délégation du contrôle de la tâche, dans un contexte d'initiative mixte, d'une manière plus sécuritaire, en s'assurant d'abord de l'égalité sémantique des points de vue et de la crédibilité de l'initiative proposée.

L'approche et le modèle théorique proposés ont été validés à l'aide d'une implantation concrète concernant un cas en commerce électronique. Un système multi-agents de marché électronique a été conçu, permettant à des agents intelligents et à un

usager de transiger des produits entre eux. Cette implantation a permis de démontrer la validité de notre approche théorique. De plus, elle a prouvé que l'utilisation judicieuse de la logique terminologique permettait de prévenir en grande partie les problèmes de compréhension liés aux conflits terminologiques.

Le calcul de la crédibilité de l'initiative mixte constitue une étape importante avant la délégation de la tâche. L'utilisation de cette technique permet non seulement le développement d'agents capables d'agir et de coopérer avec un usager d'une façon plus sécuritaire dans un contexte dynamique, tel le commerce électronique, mais aussi de trouver un équilibre entre les concepts divergents que sont l'autonomie et le contrôle, permettant ainsi de répondre à la problématique énoncée en introduction du chapitre. L'approche proposée peut également être appliquée à d'autres classes de problèmes, tels que la production automatisée et le pilotage automatique d'avions.

CHAPITRE 5

CONCLUSION GÉNÉRALE

Le travail réalisé lors de cette recherche se veut une réponse à la problématique énoncée en introduction de ce mémoire qui affirme que l'une des questions majeures concernant la cohabitation entre un utilisateur et un agent agissant en son nom est due aux divergences terminologiques entre leurs points de vue. Elle stipule que cette cohabitation hybride implique la nécessité d'une compréhension mutuelle de la part de ces deux entités à travers leurs capacités à comparer leurs opinions respectives, avant la délégation d'une tâche commune.

Plus précisément, ce mémoire a contribué à l'avancement du domaine de la coopération usager-agent par la réalisation de trois objectifs que nous nous sommes fixé. Notre premier objectif consistait à redéfinir en termes formels la comparaison des points de vue d'un usager et de son agent en s'appuyant sur la relation de subsomption de la logique terminologique. Cet objectif a été atteint grâce à l'élaboration d'une méthode judicieuse de représentation des points de vue basée sur cette logique. La réalisation de ce but a permis de démontrer que notre approche permettait, d'une part, de garantir l'égalité sémantique de ces deux points de vue avant de mesurer leurs similarités et, d'autre part, de prévenir les conflits terminologiques. Comme deuxième objectif, nous avons visé d'étendre la relation de subsomption vers une relation pondérée par des heuristiques permettant de mesurer la crédibilité de l'initiative d'un agent ou d'un usager, en vue de garantir une prise sécuritaire du contrôle de la tâche commune. Ce but a été atteint, non seulement grâce aux résultats du premier objectif, mais aussi à l'élaboration d'heuristiques permettant l'évaluation de la crédibilité d'une initiative mixte. Le troisième et dernier objectif consistait à valider notre

approche théorique par un cas concret en commerce électronique. Cet objectif a été atteint à travers le développement d'une maquette de système de ventes et d'achats de produits sur un serveur de e-commerce. La réalisation de l'ensemble de ces objectifs a été effectuée à l'aide d'une méthodologie en trois étapes.

Dans un premier temps, nous avons effectué une revue littéraire concernant la coopération dans les systèmes mutli-agents. Cela a été possible grâce à une familiarisation aux notions d'agent et de système multi-agents, notamment dans le cadre d'un cours de maîtrise sur l'intelligence artificielle dispensé par mon directeur de recherche, et aussi à travers les différentes discussions que j'ai eues avec lui ainsi qu'à la lecture de plusieurs articles reliés aux travaux sur la coopération, tels que les plans partagés [Crosz et Kraus, 1998], les intentions conjointes [Jennings, 1992], la reconnaissance de plans [Carberry, 2001] et la planification par initiative mixte [Rich et al., 2000].

Dans une seconde étape, nous avons réalisé une recherche détaillée sur la logique terminologique. Cette étude a été faite en passant d'abord en revue la notion de connaissance, suivie d'une recherche concernant l'aspect théorique et les éléments de base de la logique terminologique, pour finalement terminer avec une comparaison des SRCT disponibles à l'heure actuelle. Durant cette phase de recherche, nous avons dû réaliser plusieurs tests avec différents SRCT (PowerLoom [PowerLoom, 1997], Classic [Resnick et al., 1995]) et communiquer avec certains chercheurs du domaine afin d'obtenir des

informations, notamment avec M. Thomas A. Russ et M. Hans Chalupsky, les concepteurs du SRCT PowerLoom.

Ces phases d'investigations nous ont permis de conclure que l'ensemble des méthodes de coopération proposées utilise la théorie des interactions rationnelles, basée sur la logique des mondes possibles et ses extensions [Jennings, 1993a] [Wooldridge, 2000], pour faire communiquer un état mental à travers un acte de communication décrit par un langage tel que KQML [Finin et al., 1994]. Ces méthodes ne prennent pas en considérations les problèmes énoncés précédemment. Par ailleurs, afin de faciliter le raisonnement mutuel et de minimiser les communications, il est plus efficace d'utiliser la logique terminologique, qui se prête mieux au contexte de cohabitation, afin de transformer le problème de divergence entre deux points de vue en un problème de classification terminologique probabilisée. La définition d'un tel modèle permet d'augmenter la confiance mutuelle entre un usager et son agent, dans une perspective d'intégrer l'utilisateur et l'agent dans la même boucle de réalisation d'une tâche commune.

Brièvement, notre approche consiste à utiliser le formalisme de la logique terminologique comme support théorique pour comparer un concept d'un point de vue avec l'ensemble des concepts d'un autre point de vue, en utilisant les constructeurs et les opérateurs de subsomption offerts par le moteur d'inférence de PowerLoom. Une fois que l'égalité sémantique entre deux concepts est garantie par une relation de subsomption, nous utilisons les heuristiques que nous avons définies afin de mesurer de manière objective la

similarité entre ces deux concepts. La similarité entre deux points de vue n'est qu'une agrégation des similarités entre les concepts qui les constituent. Pour valider cette approche, nous avons créé une maquette de système de ventes et d'achats de produits sur un serveur de e-commerce IBM WebSphere [WebSphere, 2003], permettant d'héberger des agents dotés d'une base de connaissances qui peuvent transiger entre eux et se créer un point de vue de la situation du marché. Notre système permet à l'utilisateur, à travers l'interface de l'application client, de se créer un agent sur le marché, de construire son point de vue de la situation et de le comparer à celui de l'agent.

Cette approche comporte des inconvénients attribuables aux limites de la logique terminologique. Par exemple, certaines relations de subsomption peuvent ne pas être détectées parce que le nombre d'opérateurs disponibles restreint l'expressivité. L'un des objectifs futurs découlant de cette recherche est d'ailleurs de définir un langage de description des points de vue en ajoutant de nouveaux constructeurs terminologiques.

Ce mémoire ne constitue donc pas une réponse ferme et définitive à la problématique énoncée en introduction, cependant l'approche proposée doit être considérée comme un pas en avant contribuant à l'évolution du domaine de la coopération dans les systèmes multi-agents, basée sur une logique différente de celle des mondes possibles. Par ailleurs, il faut noter que nous sommes les premiers à proposer l'utilisation de la logique terminologique comme support pour la représentation et la comparaison des points de vue de l'usager et de son agent.

Dans une perspective pratique, notre approche théorique pourra être réutilisée dans d'autres classes de problèmes, tels que la production automatisée et le pilotage automatique d'avions, où la cause majeure d'accidents est le résultat d'une divergence de points de vue entre le pilote et le pilote automatique [Kitano, 1996].

Dans une autre perspective, ce mémoire ouvre la voie au développement d'une technique permettant de mesurer formellement la confiance entre un usager et son agent. Cette technique pourrait utiliser la mesure de la similarité des points de vue comme une variable pondérée qui, en conjonction avec d'autres facteurs, tels la sécurité et le niveau de risque de la tâche, pourraient servir à considérer la confiance comme une notion mesurable. D'ailleurs, des travaux en ce sens sont déjà en cours de réalisation à l'Université du Québec à Chicoutimi, sous la direction de M. Abdenour Bouzouane.

À moyen terme, ce travail de recherche sera utilisé pour l'élaboration d'une méthode permettant la fusion de deux points de vue divergents, dont l'égalité sémantique a été démontrée par notre approche, pour créer un nouveau point de vue commun respectant les règles d'agrégation, de différence et de généralisation terminologiques, afin de mettre en équivalence logique ces deux points de vue. Éventuellement, le développement d'une telle méthode pourra faire l'objet de mes recherches de doctorat, en continuité avec le présent travail.

En définitive, cette recherche, ainsi que tous les travaux qui y sont reliés, pourraient mener au développement d'une nouvelle technologie d'agents qui prendrait en charge les considérations de « l'usager dans la boucle » [Dautenhahn, 2001]. Ce travail a d'ailleurs donné lieu à une publication lors d'une conférence internationale *(IEEE-IAT'03)* [Bouzouane et Bouchard, 2003]. Certains de nos résultats jettent une lumière tout à fait nouvelle sur les questions que nous avons soulevées et qui sont reconnues comme étant importantes.

D'un point de vue personnel, ce travail m'a permis de m'initier à la recherche scientifique. Plus exactement, je me suis familiarisé au domaine de la coopération dans les systèmes multi-agents ainsi qu'à la logique terminologique. Tout au long des différentes étapes de réalisation, j'ai pu acquérir des connaissances qui me seront sûrement utiles durant ma future carrière. Je dois également souligner que cette expérience enrichissante me donne envie de continuer dans cette voie. Je trouve très gratifiant le fait d'avoir le privilège de pouvoir contribuer à l'amélioration des technologies de pointe dans le domaine de l'intelligence artificielle distribuée.

BIBLIOGRAPHIE

[Allen et Perrault., 1980] Allen J. et Perrault C.R., Analysing Intention in Utterances, in *Artificial Intelligence 15*, 1980, pages 143-178.

[Ardisonno et Sestero, 1996] Ardisonno L. et Sestero D., Using Dynamic User Models in the Recognition of the Plans of the User, in *User Modeling and User-Adapted Interaction*, 5(2), 1996, pages 157-190.

[Arens et al., 1993] Arens Y., Chee C.Y., Hsu C.N. et Knoblock C.A., Retrieving and integrating data from multiple information sources, in *International Journal of Intelligent and Cooperative Information Systems*, 2(2), 1993, pages 127-158.

[Ashley et Aleven, 1994] Ashley K.D. et Aleven V., A logical representation for relevance criteria, in *Proc. of the 1st European WorkShop on Topics in Case-Based Reasoning, Kaiserslautern, Germany, Lecture Notes in Artificial Intelligence 837*, S. Wess, K.D. Althoff et M.M. Richter éditeurs, Springer-Verlag, Berlin, 1994, pages 338-352.

[Baader, 1999] Baader F., Artificial Intelligence Today: Recent Trend and Developments, chapter Logic Based Knowledge Representation, in *Lecture Notes in Computer Science*. Springer Verlag, M.J. wooldridge et M. veloso éditeurs, 1999, pages 13-41.

[Baader et Hollunder, 1991] Baader F., Hollunder B., A Terminological Knowledge Representation System with Complete Inference Algorithms, in *Proc. of the first International Workshop on Processing Declarative Knowledge (PDK-91), Lecture Notes in Artificial Intelligence*, éditions Springer-Verlag, vol.567, 1991, pages 67-86

[Baader et al., 1991] Baader F., Bürckert H.J., Heinsohn J., Hollunder B., Müller J. Nebel B., Nutt W. et Profitlich H.J., Terminological Knowledge Representation : A proposal for Terminological Logic. Rapport technique TM-90-04 de la DFKI (Deutshes Forshungszentrum für Künstliche Intelligenz), Allemagne, 1991.

[Baader et al., 2003] Baader F., Calvanese D., McGuiness D., Nardi D. et Patel-Schneider P., The Description Logic Handbook : Theorie, Implementation, and applications, Cambridge University Press, United Kingdom, 2003.

[Bauer, 1996] Bauer M., Acquisition of User Preferences for Plan Recognition, in *Proc. of the 5th International Conference on User Modeling*, 1996, pages 105-112.

[Borgida, 1996] Borgida A., On the relative expressiveness of description logics and predicate logics, Technical Report 9600004-5, Rutgers University, New Brunswick, USA, janvier 1996.

[Bouron, 1992] Bouron T., Structures de communication et d'organisation pour la coopération dans un univers multi-agents, Thèse de l'université de Paris, 1992.

[Bouzouane et Bouchard, 2003] Bouzouane A. et Bouchard B., Human-Agent Viewpoint Similarity based on Terminological Logic for Mixed-Initiative Planning, in *Proc. of the 3th International Conference on Intelligent Agent Technology (IEEE-IAT'03)*, Halifax, Canada, 13-17 Octobre 2003, pages 1-5.

[Brachman, 1977] Brachman R.J. et Levesque H., Pigman V. An essential hybrid reasoning system : Knowledge and symbol level accounts of krypton, in *IJCAI-85*, 1985, pages 532-539.

[Brachman et Schmolze, 1985] Brachman R.J. et Schmolze J.G., An overview of the KL-ONE knowledge representation system, in *Cognitive Science*, 9(2), 1995, pages 171-216.

[Carberry, 2001] Carberry S., Techniques for Plan Recognition, User Modeling and User Adapted-Interaction, vol. 11, 2001, pages 31-48.

[Carberry et Lambert, 1999] Carberry S. et Lambert L., A Process Model for Recognizing Communicative Acts and Modeling Negotiation Subdialogues, in *Computational Linguistics*, 25(1), pages 1-53.

[Castelfranchi, 2000] Castelfranchi C., Conflicts ontology, Computational conflicts, in *Computational conflicts – Conflicts Modeling for Distributed Intelligent Systems*, Müller, H.J. et Dieng R. éditeurs, 2000, pages 21-40.

[Castelfranchi et Falcone, 1997] Castelfranchi C., Falcone R., Delegation Conflicts, in *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 1997, pages 234-254.

[Castelfranchi et Falcone, 1998] Castelfranchi C., Falcone R., Towards a Theory of Delegation for Agent-Based Systems, in *Robotics and Autonomous Systems (Multi-Agent Rationality)*, 24(3), 1998, pages 141-157.

[Castelfranchi et Falcone, 2000] Castelfranchi C., Falcone R., Trust and Control : A Dialectic Link, in *Applied Artificial Intelligence Journal*, 14(8), 2000, pages 799-823.

[Cesta et D'Aloisi, 1999] Cesta A., D'Aloisi D., Mixed Initiative Issues in an Agent-Based Meeting Scheduler, in *User Modeling and User-Adapted Interaction Journal*, 9(2), 1999, pages 45-78.

[Chaib-Draa et al., 1992] Chaib-Draa B., Moulin B., Mandiau R. et P. Millot, Trends in distributed artificial intelligence, in *Artificial Intelligence Review*, 6(1), 1992, pages 35-66.

[Chaudron et al., 2000] Chaudron L., Fiorino H., Maille N. et Tessier C., Difference : a key to enrich knowledge. Concepts and models, Müller H. J. et Dieng R. éditeurs, in *Computational conflicts – Conflict Modeling for distributed Intelligent Systems*, 2000, pages 82-102.

[Chavez et Pattie, 1996] Chavez A. et Pattie M., Kasbah: An Agent Marketplace for Buying and Selling Goods, In *Proc. of the First Int'l Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, 1996, pages 75-90.

[Ciancarini et al., 1999] Ciancarini P., Omicini A. et Zambonelli F., Coordination Models for Multi-Agent Systems, in *Agentlink News Journal*, 3, 1999, pages 3-6.

[Cohen et Levesque, 1991] Cohen P. R. et Levesque H. J., Teamwork, Nous, 25(4), 1991, pages 487-512.

[Cohen et al., 1991] Cohen P. R., Song B., Spencer B. et Beek P., Exploiting Temporal and Novel Information from the User in Plan recognition, in *User Modeling and User-Adapted Interaction*, 1(2), 1991, pages 125-148.

[Conte et al., 1991] Conte R., Miceli M. et Castelfranci C., Limits and Levels of Cooperation : Desentangling Various Types of Prosocial Interaction, in *Distributed A.I. 2*, Demazeau Y. et Müller J.-P éditeurs, North-Holland, 1991.

[Chu et al., 1993] Chu W.W., Merzbacher M.A., et Berkovich L., The design and implementation of CoBase, in *Proc. of the ACM/SIGMOD International Conference on the Management of Data, Washinton (DC), USA*, ACM SIGMOD Record, 22(2), 1993, pages 517-522.

[Dautenhahn, 2001] Dautenhahn K., Socially Intelligent Agents: The Human in the loop, in *IEEE Trans on systems*, 31(5), 2001, pages 345-348.

[Davies et al., 2003] Davies J., Van Harmelen F. et Fensel Dieter, Towards the semantic web: ontology-driven knowledge management, éditions Chichester, England, Hoboken, 2003, pages 1-288.

[Davis et Smith, 1983] Davis R. et Smith R.J., Negotiation as a Metaphor for Distributed Problem Solving, in *Artificial Intelligence*, 20(1), 1983, pages 63-109.

[Deitel et Deitel, 2000] Deitel H.M. et Deitel P.J., Comment Programmer en Java, éditions Reynald Goulet inc., 2001, pages 1184-1289.

[Demazeau et Müller, 1991] Demazeau Y. et Müller J.-P, Decentralized Artificial Intelligence 2., Elsevier North-Holland, 1991.

[Devanbu et Litman, 1991] Devanbu P.T. et Litman D.J., Plan-based terminological reasoning, in *Proc. of the second International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, Cambridge, 1991, pages 128-138.

[Diday et al., 2000] Diday E., Kodratoff Y., Brito P. et Moulet M., Induction symbolique numérique à partir de données, éditions Cépaduès, Toulouse, France, 2000, pages 168-196.

[Doran et al., 1997] Doran J.E., Franklin S., Jennings N.R., et Norman T.J., On cooperation in multi-agent systems, in *The Knowledge Engineering Review*, 12(3), 1997, pages 309-314

[Durfee et Lesser, 1991] Durfee E. etLesser V.R., Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation, in *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5), 1991, pages 1167-1183.

[Falcone et Castelfranchi, 2001] Falcone R. et Castelfranchi C, The Human in the loop of Delegated Agent : The Theory of Adjustable Social Autonomy, in *IEEE Trans*, 33(5), 2001, pages 406-418.

[Farreny et Ghallab, 1987] Farreny H. et Ghallab Malik, Éléments d'intelligence artificielle, *Traité des Nouvelles Technologies série Intelligence Artificielle*, éditions Hermès, Paris, 1987, pages 31-69.

[Ferber, 1995] Ferber J., Les systèmes multi-agents : vers une intelligence collective, éditions InterEditions, Paris, 1995, pages 6-95.

[Finin, 1986] Finin T.W., Interactive Classification: A technique for acquiring and maintaining Knowledge Bases, in *Proc. IEEE*, vol. 74, 1986, pages 1414-1421.

[Finin et al., 1994] Finin T., Fritzson R., McKay et McEntire R., KQML as an Agent Communication Language, in *Proc. Third International Conference on Information and Knowledge Management (CIKM '94)*, 1994.

[Galliers, 1991] Galliers J. R., Modelling Autonomous Belief Revision in Dialogue, in *Decentralized Artificial intelligence 2 : Proc. Of the Second European Workshop on Autonomous Agents in a Multi-Agents World (MAAMAW'90)*, Demazeau Y. et Müller J.-P. éditeurs, Elsevier North-Holland, 1991.

[Grosz et Kraus, 1993] Grosz B.J. et Kraus S., Collaborative plans for group activities, in *Proc. of the 1993 International Joint Conference on Artificial Intelligence (IJCAI-1993)*, San Mateo, 1993, pages 367-373.

[Grosz et Kraus, 1996] Grosz B.J. et Kraus S., Collaborative plans for complex group action, in *Artificial Intelligence*, 86(2), 1996, pages 269-357.

[Grosz et Kraus, 1998] Grosz B.J. et Kraus S., The evolution of SharedPlans, Wooldridge M. et Rao A. éditeurs, in *Foundations and Theories about Rational Agency*, 1998, pages 1-31.

[Hadad et Kraus, 1999] Hadad M. et Kraus S., SharedPlans in electronic commerce, in *Intelligent Information Agents*, Matthias Kusch éditeurs, Springer-Verlag, Berlin, 1999.

[Haddadi, 1996] Haddadi A., Communication and Cooperation in Agent Systems, Springer-Verlag, 1996.

[Hall, 1962] Hall Arthur D., A methodology for Systems Engineering, Princeton, NJ: Van Nostrand, 1962.

[Hamburger et Richards, 2002] Hamburger H. et Richards D., Logic and Language Models for Computer Science, éditions Prentice-Hall, Upper Saddle River, 2002.

[Haton et al., 1991] Haton J-P., Bouzid N., Charpillet F., Haton M., Lâasri B., Lâarsi H., Marquis P., Mondot T. et Napoli A., Le raisonnement en intelligence artificielle, éditions InterEditions, Paris, 1991.

[Hayes, 1980] Hayes P., The Logic of Frames, in *Frame Conceptions and Text Understanding*, éditions Gruyter, 1980, pages 46-61.

[Hearst et al., 1999] Hearst M., Allen J.F., Guinn C.I., Horvitz E., Mixed-initiative Interaction, in *Intelligent Systems, IEEE Computer Society*, 1999, pages 14-24.

[Jennings, 1992] Jennings N.R., Joint Intentions as a Model of Multi-Agent Cooperation in Complex Dynamic Environments, Thèse de doctorat, University of London, 1992, pages 1-179.

[Jennings, 1993a] Jennings N.R., Commitments and conventions : the foundation of coordination in multi-agents systems, in *The Knowledge engineering Review*, 8(3), 1993, pages 223-250.

[Jennings 1993b] Jennings N.R., Specification and implementation of a Belief-Desire-Joint-Intention architecture for collaborative problem solving, in *Journal of Intelligent and Cooperative Information System*, 2(3), 1993, page 289-318.

[Jennings et al., 1998] Jennings N.R., Parsons S., Noriega P. et Sierra C., On argumentation-based negociation, in *Proc. Of the International Workshop on Multi-Agents Systems*, 1998.

[Jung et al., 2001] Jung H., Tambe M., Kulkarni S., Argumentation as Distributed Constraint Satisfaction : Application and results, in *Proc. Fifth Conference on Autonomous Agents*, ACM Press, 2001.

[Kitano, 1996] Kitano H., New Directions, Nausicaä and the Sirens, A Tale of Two Intelligent Autonomous Agents, in *IEEE Computer Society*, 11(6), 1996.

[Kraus et al., 1998] Kraus S., Sycara K. et Evenchik A., Reaching agreements through argumentation : a logical model and implementation, in *Artificial Intelligence*, 1998.

[Larousse, 1980] Larousse, Dictionnaire encyclopédique pour tous: le petit Larousse illustré, librairie Larousse, Paris, 1980, page 948.

[Lesh, 1997] Lesh N., Adaptive Plan Recognition, in *Proc. of the 15th International Joint Conference on Artificial Intelligence*, 1997, pages 1204-1208.

[Lesh et al., 1999] Lesh N., Rich C et Sidner C.L., Using Plan Recognition in Human-Computer Collaboration, in *Proc. Seventh Conference on User Modeling*, Banff, Canada, 1999, pages 1-11.

[Levesque et Brachman, 1987] Levesque H. et Brachman R.J., Expressiveness an Tractability, in *Knowledge Representation and Reasoning , Computational Intelligence*, vol. 3, 1987, pages 78-93.

[Lipkis, 1982] Lipkis T., A KL-ONE Classifier, in *Proc. of KL-ONE Workshop*, Jackson, Mississippi, 1982, pages 126-143.

[Lochbaum et al., 1991] Lochbaum K.E., Grosz B.J. et Sidner C.L., Models of Plans to Support Communication, in *Proc. of the 8th National Conference on AI*, Boston, USA, 1990, pages 485-490.
[Lochbaum, 1994] Lochbaum K.E., The Need for Intentionnaly-Based Approaches to Language, Aiken Computation Lab, Havard University, Cambridge, 1994, pages 1-4.

[LOOM, 1991] LOOM users guide (version 1.4), Information Science Institute, University of Southern California, Marina de Rey (CA), USA, 1991.

[Luck et al., 1987] Luck K., Nebel B., Peltason C. et Schmiedel A., The anatomy of the back-system. Technical report, Department of Computer Science, Technische Universität Berlin, Belin, Germany, 1987.

[McCarthy et Hayes, 1969] McCarthy J. et Hayes P.J., Some Philosophical Problems from the Standpoint of Artificial Intelligence, in *Machine intelligence 4*, Edinburgh University Press, Edimbourg, 1969.

[McGuinness et al., 2002] McGuinness D., Smith M., Volz R., et Welty C., Web Ontology Language (OWL) Guide version 1.0, Document préliminaire du consortium W3C, http://www.w3.org/tr/2002/WD-owl-guide-20021104/, 4 novembre 2002.

[McGregor, 1988] McGregor R., A deductive Pattern Matcher, in *Proc. 7th AAAI*, Saint-Paul, Minnesota, 1988, pages 403-408.

[McGregor, 1991] McGregor R., The evolving technology of classification-based knowledge representation systems, in *Principles of Semantic Networks : Explorations in the Representation of Knowledge*, Morgan Kaufmann Publishers, San Mateo, USA, 1991, pages 385-400.

[MacGregor et Burstein, 1991] MacGregor R. et Burstein M.H., Using a description classifier to enhance knowledge representation, in *IEEE Expert*, 6(3), 1991, pages 41-46.

[McGregor et Brill, 1992] MacGregor R. Brill D., Recognition algorithms for the looms classifier, in *AAAI-92*, MIT Press, 1992, pages 774-779.

[Minsky, 1974] Minsky M., A Framework for Representing Knowledge, Mémo technique 306 du laboratoire d'intelligence artificielle du MIT, Juin 1974.

[Napoli, 1997] Napoli A., Une introduction aux logiques de descriptions, in *Projet SYSCO*, Rapport de recherche no.3314, 1997, pages 7-50.

[Nebel, 1995] Nebel B., Reasoning and Revision in Hybrid Representation Systems, in *Lecture Notes in Artificial Intelligence*, éditions Springer-Verlag, vol. 422, 1995.

[Nilsson, 1998] Nilsson N., Artificial Intelligence : A New Synthesis, Morgan Kaufmann éditeurs, 1998.

[Norman, 1994] Norman D., How Might People Interact with Agent, in *Communication of the ACM*, 37(7), 1994, pages 68-71.

[Penders, 1999] Penders J., The Pratical Art of Moving Physical Objects, Thèse de doctorat, University of Maastricht, 1999.

[PowerLoom, 1997] PowerLoom Manual (version 1.0), Information Science Institute, University of Southern California, Marina de Rey (CA), USA, 1991.
[Pylyshyn, 1989] Pylyshyn Z., Computing in Cognitive Science, in *Foundations of Cognitive Science,* Postner M.I. éditeurs, Cambridge, MIT press, 1989, pages 51-91.

[Raskutti et Zukerman, 1991] Raskutti B. et Zukerman I., Generation and selection of likely interpretations during plan recognition in task-oriented consultation systems, in *User Modeling and User-Adapted Interaction*, 1(4), 1991, pages 323-353.

[Resnick et al., 1995] Resnick L.A., Borgida A., Brachman R.J., McGuinness D.L, Patel-Schneider P. et Zalondek K.C., CLASSIC. Description and Reference Manual for Common Lisp Implementation (Version 2.3), AT&T Bell Laboratories, Murray Hill (NJ), USA, 1995.

[Rich et al, 2000] Rich C., Sidner L. et Lesh N., COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction, in *AI magazine*, Mitsubichi Electric Research Laboratories, Cambridge, 2000, pages 1-12.

[Rao et Georgeff, 1991] Rao A. et Georgeff P., Modeling Rational Agents within a BDI-Architecture, in *Proc. of the second International Conference on principles of Knowledge Representation and Reasoning (KR91)*, San Mateo, 1991, pages 1-18.

[Rao et Georgeff, 1995] Rao A. et Georgeff P., BDI Agents: From Theory to Practice, in *Proc. of the first International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, USA, 1991, pages 1-12.

[Rao et al., 1992] Rao A., Georgeff P. et Sonenberg E., Social Plans : A Preliminary Report, in *Decentralised AI 3*, North Holland, 1992, pages 57,76.

[Schmidt, 1991] Schmidt R., Algebraic Terminological Representation, Rapport technique MPI-I-91-216 de MPI (Max-Planck-Institut für informatik), Allemagne, 1991.

[Schmidt, 1992] Schmidt R., Terminological Representation, Natural Language and Relation Algebra. Rapport technique MPI-I-92-246 de MPI (Max-Planck-Institut für informatik), Allemagne, 1992.

[Schmidt, 2000] Schmidt R., Relational Grammars for Knowledge Representation, in *Variable-Free Semantics*, collection Artikulation und Sprache, éditions Secolo Verlag, vol. 3, 2000, pages 162-180.

[Schmiedel, 1992] Schmiedel A., For a more expressive query language, in *AAAI fall Symposium Series – Issues in Description Logics : Users Meets Developers*, Cambridge (MA), USA, 1992, pages 98-102.

[Schmolze et Lipkis, 1983] Schmolze J.G. et Lipkis T.A., Classification in the KL-ONE Knowledge Representation System, in *Proc. of the 8th IJCAI*, Karlsruhe, 1983, pages 330-332.

[Searle, 1991] Searle J., Collective Intentions and Actions, in *Intentions in Communication*, Cohen P.R. et Morgan J. éditeurs, MIT press, 1991, pages 401-416.

[Sowa, 1984] Sowa J., Conceptual structures, éditions Addison-Wesley, 1984.

[Spaan, 2002] Spaan M.T.J., Team play among soccer robots, Thèse de maîtrise, University of Amsterdam, 2002.

[Tambe, 1997] Tambe M., Towards flexible teamworks, in *Journal of Artificial Intelligence Research (JAIR)*, vol. 7, 1997, pages 83-124.

[Tambe et al., 1997] Tambe M., Adibi J., Al-Onaizan Y., Kaminka A., Marsella S., Muslea I. Et Tallis M., ISIS : Using an Explicit Model of Teamwork in Robocup'97, in *Proc. of the RoboCup'97 simulation league tournament*, Information Sciences Institute, University of Southern California, 1997, pages 1-7.

[Tambe et Jung, 1999] Tambe M. et Jung H., The benefits of arguing in a team, in *AI magazine*, 20(4), 1999.

[Tessier et Chaudron, 1998] Tessier C. et Chaudron L., Conflicts among agents : avoid or use them?, in *Working notes of the ECAI-98 Workshop*, Tessier C. et Chaudron L. éditeurs, 1998.

[Tessier et al., 2001] Tessier C., Chaudron L., Müller H-J., Conflicting agents : Conflict Management In Multi-Agent systems, Kluwer Academic Publishers, 2001, pages 1-30.

[Tuomas, 1997] Tuomas A., On the Structure of Delegation Networks, in *Proc. of The 11th Computer Security Foundations Workshop*, 1997, pages 1-13.

[Valancia, 2000] Valentia E., Outils de topologie algébrique pour la gestion de l'hétérogénéité sémantique entre agents dialogiques, Thèse de Doctorat, Université paris-XI, 2000, pages 37-72.

[Watzlawick et al., 1967] Watzlawick P., Helmick Beavin J. et Jackson D.D., Pragmatics of human communication. A study of interactional patterns, pathologies, and paradoxes, New York, 1967.

[WebSphere, 2003] WebSphere, IBM WebSphere Technology for E-Commerce, http://www.ibm.com/us/, 2003.

[Weida, 1993] Weida R., Terminological Constraint Network Reasoning and its Application to Plan Recognition, Thèse de doctorat, University of Columbia, 1993, pages 9-34.

[Weida et Litman, 1992] Weida R. et Litman D.J., Terminological reasoning with constraint networks and an application to plan recognition, *Proc. of the 3th International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, 1992, pages 282-293.

[Werner, 1990] Werner E., Cooperating agents: a unified theory of communication and social structure, in *Distributed Artificial Intelligence: II*, Cohen P., Morgan J. et Pollack M. éditeurs, 1990.

[Wilensky, 1978] Wilensky R., Why John Married Mary: Understanding Stories Involving Recurring Goals, in *Cognitive Science 2*, 1978, pages 235-266.

[Wooldridge, 2000] Wooldridge M., Reasoning about rational agents, MIT Press, Cambridge, Massachusetts, London, England, 2000.

[Wooldridge et Jennings, 1994] Wooldridge M. et Jennings N.R., Agent Theories, Architectures, and Languages: A Survey, in *Proc. of Workshop on Agent Theories, Architectures and Languages (ECAI94)*, Springer-Verlag, Amsterdam, 1994, pages 1-32.

[Wooldridge et Jennings, 1995] Wooldridge M. et Jennings N., Intelligent agents, theory and practice, in *Knowledge Engineering Review*, 10(2), 1995.

[Wu, 1991] Wu D., Active Acquisition of User Models: Implication for Decision-Theoric Dialog Planning and Plan Recognition, in *User Modeling and User-Adapted Interaction*, 1(2), 1991, pages 149-172.

[Yen et al., 1991a] Yen J., Juang H.-L., et MacGregor R., Using polymorphism to improve expert system maintainability, in *IEEE expert*, 6(2), 1991, pages 48-55.

[Yen et al., 1991b] Yen J., Neches R., et MacGregor R., CLASP : Integrating term subsumption systems and production systems, in *IEEE Transactions on Knowledge and Data Engineering*, 3(1), 1991, pages 25-32.

[Yokoo et Hirayama, 1998] Yokoo M. et Hirayama K., Distributed constraint satisfaction algorithm for complex local problems, in *Proc. Of the 3rd Intl. Conf. On Multi-Agent Systems*, Juin 1998.

ANNEXE 1

EXEMPE DE FICHER DE TRADUCTION GÉNÉRÉ PAR LE TRADUCTEUR DE POINTS DE VUE

```
;;; ******************************************************
;;; *    Fichier généré par le programme Cohabitation    *
;;; *          Traduction des PV en LT                   *
;;; ******************************************************

;;; Initialisation de la base de connaissance
;;; ******************************************************

(in-package "STELLA")
(defmodule "PL-USER/COHABITATION")
(in-module "PL-USER/COHABITATION")
(clear-module "PL-USER/COHABITATION")
(reset-features)
(in-dialect :KIF)

;;; Définition des concepts et des rôles
;;; ******************************************************

(defconcept VP (?x))
(defconcept PLAN (?x))
(defconcept BELIEF (?x))
(defconcept GOAL (?x))

(defconcept VPH (?x VP):<=> (exists (?p ?b ?g)
                          (and (BELIEFh ?b)
                               (PLANh ?p)
                               (GOALh ?g))))

(defconcept VPA (?x VP):<=> (exists (?p ?b ?g)
                          (and (BELIEFa ?b)
                               (PLANa ?p)
                               (GOALa ?g))))

(defrelation Purchasing-Product ((?x GOALh) (?y Product)):<=> (Product
?y))

(defconcept Product (?x))

(defconcept GOALh (?x GOAL):<=> (exists (?a )
(and (Purchasing-Product ?x ?a))))

(defrelation Contact-Supplier ((?x PLANh) (?y Supplier)):<=> (Supplier
?y))

(defconcept Supplier (?x))

(defrelation Determine-Date-Supply ((?x PLANh) (?y DateSupply)):<=>
(DateSupply ?y))

(defconcept DateSupply (?x))

(defconcept PLANh (?x PLAN):<=> (exists (?a ?b )
(and (Contact-Supplier ?x ?a)
```

```
     (Determine-Date-Supply ?x ?b))))

(defrelation Missing-Product ((?x BELIEFh) (?y Product)):<=> (Product
?y))

(defconcept BELIEFh (?x BELIEF):<=> (exists (?a )
(and (Missing-Product ?x ?a))))

(defconcept GOALa (?x GOAL):<=> (exists (?a )
(and (Purchasing-Product ?x ?a))))

(defrelation Find-Wholesaler ((?x PLANa) (?y Wholesaler)):<=> (Wholesaler
?y))

(defconcept Wholesaler (?x))

(defconcept PLANa (?x PLAN):<=> (exists (?a ?b )
(and (Find-Wholesaler ?x ?a)
     (Determine-Date-Supply ?x ?b))))

(defconcept BELIEFa (?x BELIEF):<=> (exists (?a )
(and (Missing-Product ?x ?a))))

;;; Définition des relations de subsomption pré-établies
;;; ****************************************************

(assert (subset-of Supplier Wholesaler))
```