

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université SAAD DAHLEB de BLIDA

Faculté des sciences

Département d'informatique



MEMOIRE

Pour l'obtention du diplôme d'Ingénieur d'Etat en Informatique

Option : systèmes d'information avancés

Thème

**Un système multi-agents pour la coopération de
ressources dans un SI coopératif**

Proposé et suivi par

Mme : L. OUAHRANI

Réalisé par :

Mlle : DJELLAL SARA

Mlle : DRAMSSAOUD FAIZA

PROMOTION

2006/2007

MIG-004-167-1

Remerciements

En témoignage de nos profonds sentiments de respect et de reconnaissance, nous tenons à remercier Madame OUAHRANI pour avoir proposé un sujet dans un domaine intéressant, comme

*Les systèmes multi agents pour la coopération des SI,
et pour ses directives, ses recommandations, ses remarques constructives et sa disponibilité.*

Nos remerciements vont également aux membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'examiner ce travail.

Dédicace

C'est avec mes sentiments de joie et de bonheur que je dédie cette thèse à Mes chers parents pour leurs amour, aide, sacrifice, et pour leur soutient tout au long de mon existence. Ainsi qu'à :

- *Mes grands parents.*
- *Mes chers sœurs : MOUNIA et LIDIA.*
- *Ma famille.*
- *Mon binôme FAIZA.*
- *Tous ceux qui m'ont aidé, soutenu et encouragé à aller vers l'avant et à achever ce travail.*
- *Toutes mes amies.*
- *Ainsi qu'à tous ceux qui m'aiment et que j'aime.*

DJELLAL SARA

Dédicace

C'est avec mes sentiments de joie et de sincérité que je dédie cette thèse à Mes chers parents pour leurs amour, aide, sacrifice, et pour leur soutien tout au long de mon existence. Ainsi qu'à :

- *Mes grands parents.*
- *Mes chers sœurs : KENZA, NADIRA, KAOUTAR.*
- *Mon cher frère ABDERAHMAN.*
- *Mes deux famille : DRAMSSAOUD et BOUCHAMA.*
- *Mon binôme SARA.*
- *Tous ceux qui m'ont aidé, soutenu et encouragé à aller vers l'avant et à achever ce travail.*
- *Toutes mes amies.*
- *Ainsi qu'à tous ceux qui m'aiment et que j'aime.*

DRAMSSAOUD FAIZA

Sommaire



Introduction générale

Première partie

Etat de l'art

Chapitre I Vers des SI coopératifs

I. Evolution de système d'information.....	4
I.1. Méthodologie.....	4
I.2. Architecture.....	5
I.2.1. <i>Architecture client/serveur</i>	5
I.2.2. <i>systèmes distribuées</i>	7
I.3. La coopération des SI.....	8
I.3.1. notion de coopération.....	9
I.3.2. Méthodes de coopération.....	11
I.3.3. Niveau de coopération.....	13
I.3.4. Les outils des systèmes coopératifs.....	14
1) <i>Le Travail Coopératif Assisté par Ordinateur (TCAO)</i>	14
2) <i>Le groupware</i>	15
3) <i>Le workflow</i>	18
4) <i>SIC à base d'agent</i>	20
II. SIC : Système d'Information Coopératif.....	21
II.1. Propriétés des Systèmes d'Information Coopératifs.....	22
II.2. Techniques de coopération.....	26
II.2.1. Les systèmes d'Information Fédérés (SIF).....	26
1) <i>Objectifs et problématique</i>	27
2) <i>Types de fédération</i>	28
II.2.2. Les systèmes de coopération à base de système multi agents.....	30
II.3. Problématique de la coopération de SI.....	30
Conclusion.....	31

Conclusion.....	31
-----------------	----

Chapitre II Les Systèmes Multi Agents

1. Historique.....	32
2. Définition d'un agent.....	33
3. Différentes catégories d'Agents.....	35
3.1 <i>Agents cognitifs</i>	35
3.2 <i>Agents réactifs</i>	37
3.3 <i>Agents hybrides</i>	38
4. Définition d'un système multi agents (SMA).....	39
4.1 Organisation	40
4.2 Les types de SMA.....	41
4.3 Formes d'allocation de tâches dans un SMA.....	42
4.4 Interaction dans un SMA.....	42
1. La communication.....	43
1.1. <i>La communication indirecte</i>	43
1.2. <i>La communication directe</i>	44
2. La coopération	49
3. La coordination.....	50
4. La négociation.....	50
5. Agent Oriented Programming.....	51
6. Les SMA et la coopération des SI.....	52
6.1. <i>Agent informationnel</i>	52
6.2. <i>Agent Informationnels pour la recherche et l'intégration de l'information..</i>	53
6.3. <i>Fonctionnalités des AI pour la recherche et l'intégration d'information.....</i>	54
6.4. <i>Principaux concepts, techniques et langage utilisés</i>	54
Conclusion.....	55

Chapitre III Langage de modélisation AUMML

1. Diagrammes de séquence AUMML.....	58
2. Diagramme d'interaction de classe.....	59
Conclusion.....	60

Deuxième partie

Conception et réalisation du SIC

Chapitre I Conception du SIC.....	61
I. Une approche unifiée du SI.....	62
I.1 Organisation du Système Multi Agents.....	64
I.2 Typologie des agents du SMA.....	65
II. La modélisation du système	67
II.1 Diagrammes de cas d'utilisation.....	68
II.1.1 Organisation des interactions dans les SMA.....	70
II.1.2 protocole d'allocation de ressources.....	72
II.2 Diagrammes de séquence.....	73
II.3 Diagrammes d'interactions de classe d'agents.....	80
II.4 Diagrammes d'activité	82
II.5 Diagrammes de collaboration.....	89
III. Etude de cas.....	93
1. Description du système	94
2. description des agents et des services	96
3. fonctionnement du système	99
4. présentation des mécanismes de coopération utilisés	102
Chapitre II Réalisation	
I. Description des outils utilisés	104
II. Creation d'un agent avec AgentBuilder.....	105
III. Réalisation.....	108
Conclusion générale	113
Bibliographie.....	115

Introduction générale

Les récents progrès réalisés en matière de communication (réseaux hauts débits, normalisation des protocoles et des architectures a objets repartis, explosion de l'Internet) permettent aujourd'hui d'envisager la construction de systèmes d'information de grande envergure au coeur desquels se trouvent de gros volumes d'information multiforme, repartis a l'échelle planétaire sur des équipements varies (de la carte a puce aux gisements de données).

Le domaine de l'informatique et plus particulièrement celui des SI a connu une profonde mutation depuis une dizaine d'années, en particulier les besoins d'ouverture et de communication entre les SI sont devenus importants. Les nouvelles applications telles que les systèmes d'aide à la décision, les systèmes d'aide au commandement, les applications de simulation et le commerce électronique nécessitent une communication entre SI.

Les applications ont besoins d'accès coordonnés à différentes sources de données afin de synthétiser les informations provenant de ces sources. Le partage des informations présente des intérêts économiques non négligeables puisqu'il permet de réduire les coûts d'acquisition et de mise à jour des informations. Ce partage nécessite également une communication inter système. Par conséquent, les SI actuels doivent désormais être coopératifs mais leurs répartitions et leurs hétérogénéités constituent des obstacles majeurs à leurs coopérations.

De nombreux travaux menés dans plusieurs domaines de recherche ont pour objectifs de proposer des solutions pour la conception et le développement de tels systèmes d'information. Ces recherches ont abouti à l'émergence de différentes techniques supportant l'intégration de SI distribués, hétérogènes et autonomes. C'est ainsi que les recherches dans le cadre de la coopération de SI tentent de tirer une partie des modèles de coopération, de comportement et de représentation de connaissances développée en Intelligence Artificielle Distribuée et dans les SMA.

L'enjeu est d'offrir une assistance intelligente aux utilisateurs et aux applications par le biais d'agents autonomes et coopérants.

Objectif :

Notre travail consiste à développer un système multi agents à savoir un ensemble d'agents logiciels communicants, pour l'intégration de ressources dans un système d'information coopératif.

Ces ressources (données et fonctions associées) sont réparties de manière dynamique sur un ensemble non figé d'équipements tout en offrant à l'utilisateur un accès transparent.

Dans ce cadre nous proposons une approche de coopération s'articulant autour d'agents Ressource et d'agents Fédérateur et disposant, à la manière des acteurs dans une entreprise distribuée, de compétences et de connaissances. L'intégration de ces agents formerait un système cohérent et fidèle de la structure de l'entreprise qui jouerait le rôle de médiateur de communication entre ces agents.

Vis-à-vis de nos préoccupations en matière de coopération de SI à base d'agent les problématiques que nous retenons sont les suivantes :

- Comment faire en sorte que l'organisation de SMA soit la projection naturelle de l'organisation de l'entreprise orientée d'avantage vers le client et sujette à une perpétuelle évolution ?
- Quelles sont la forme et la fonction concernant un agent sachant que le comportement doit dépendre de cette forme et de cette fonction ?
- Comment rechercher de façon transparente la ressource adéquate dans un environnement ouvert et dynamique ?
- Comment négocier avec le serveur pour obtenir le service le plus adapté ?
- Quelles sont les formes d'interactions permettant aux agents de trouver conjointement une réponse à une requête d'un utilisateur ?

Organisation du mémoire :

Ce mémoire est organisé en deux parties

Première partie : est organisée en trois chapitres :

L'objectif du premier chapitre est de présenter l'évolution des systèmes d'information d'un point de vue méthodologie d'analyse, architecture, échange d'information et coopération.

Nous présentons la coopération ; propriétés et techniques, pour en discuter enfin la problématique de coopération.

Le deuxième chapitre représente un état de l'art sur les SMA et leurs application dans les SI coopératifs, nous donnons une vue détaillée sur la notion d'agent et de système multi agents.

Dans le troisième chapitre nous présentons brièvement le langage de modélisation choisi AUML, en ne représentant que les extensions par rapport à l'UML.

Deuxième partie : est consacré à la conception et la réalisation du système, elle est structurée en trois chapitre :

Dans le premier chapitre nous modélisons le système avec AUML, en utilisant les diagrammes qui décrivent l'état statique et dynamique de ce système.

Nous proposons dans le deuxième chapitre une étude de cas pour illustrer le système.

le troisième chapitre est consacré à la réalisation de notre étude de cas, Nous y présentons également l'interface utilisateur que nous avons élaborée afin qu'un utilisateur humain puisse interagir avec les agent du système.

Première partie

Chapitre I : Vers des SI coopératifs

I. Evolution de système d'information :

Toute organisation est pilotée par une direction, une équipe dirigeante. Ce système de pilotage a pour mission de conduire l'organisation vers ses objectifs. Et c'est dans ce contexte qu'apparaît le système d'information. Ce sous système de l'organisation s'occupe de récolter l'information, de la stocker, de la traiter et de la diffuser dans le système de pilotage.

« Le système d'information est le système de couplage entre le système opérant et le système de pilotage » [LEM 74].

Les systèmes d'information (SI) jouent un rôle stratégique, dont dépend parfois la survie de l'entreprise, et pourtant leur implantation actuelle dans les organisations est loin de répondre aux besoins et exigences des décideurs.

Dans ce chapitre nous présentons les différentes méthodologies d'analyse d'un SI, ensuite le passage de l'architecture client /serveur vers l'architecture distribuée pour arriver à la notion de coopération.

I.1 Evolution des méthodologies d'analyse :

Les méthodes de conception de SI de la première génération (les méthodes cartésiennes) sont basées sur des concepts et techniques de décomposition hiérarchique des processus et de flux de données (flux d'information) les méthodes cartésienne préconisent d'analyser et de concevoir le SI en se centrant sur ces fonction.

Les méthodes de conception de SI de la seconde génération (les méthodes systémiques) sont entièrement centrées sur la modélisation des données et la modélisation des traitements ; le SI est un « modèle » de la réalité organisationnelle qui apporte aux acteurs et décideurs la connaissance dont ils ont besoin pour agir et décider.

On peut considérer l'approche objet comme une évolution de l'approche systémique vers une plus grande cohérence entre les objets et leurs dynamiques. C'est en effet la principale contribution de l'objet :

Décrire une grande partie de la dynamique du SI comme un ensemble d'opérations attachés aux objets composants ce système, ceci donne une meilleure compréhension de la sémantique des objets et permet une meilleure modularité et réutilisation des composants du SI [BOU 00].

L'aspect fonctionnel est très important dans les organisations considérer chaque fonction comme un objet relève d'un effort d'abstraction qui n'intéresse guère que les réalisateurs de compilateurs ou les spécialistes, dont le but est d'identifier un sous ensemble minimal de concepts à réaliser.

Certaines méthodes et les plus utilisés comme OMT conservent cette dimension fonctionnelle.

I.2. Architecture

I.2.1. Architecture client/serveur

Les architectures client/serveur sont apparues vers 1985 à la suite de la popularité grandissante des BD et des réseaux locaux. Aujourd'hui, le concept de client/serveur est souvent synonyme de poste de travail clients dotés d'interfaces graphiques permettant d'accéder à des BD stockées sur des serveurs. L'architecture illustrée à la figure 1.2 élargit le concept du client/serveur présenté dans la figure 1.1, on y découvre un réseau local fédérant plusieurs serveurs : de fichier, de BD, de fax et de messagerie.

Pour la plupart des applications et des architectures client/serveur, le réseau local fournit une structure de connectivité. Les réseaux locaux s'implantant progressivement dans les entreprises, les utilisateurs découvrent de nouvelles manières d'utiliser et de partager les données et les ressources.

Les architectures client/serveur, dans lesquelles les clients exécutent les applications frontales (via des interfaces graphique) pour accéder aux données et aux ressources fournies par les serveurs sont devenues très populaires.

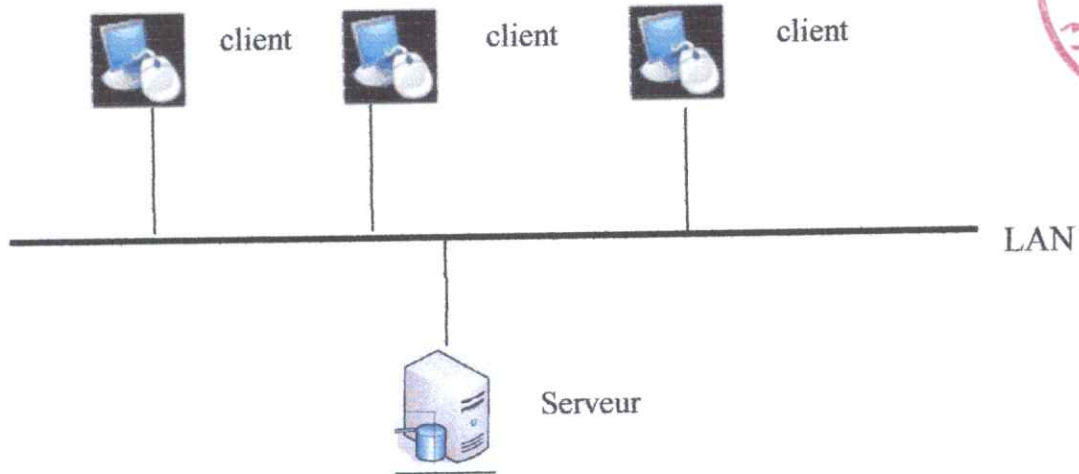


Figure 1.1 : architecture client/serveur sur un réseau

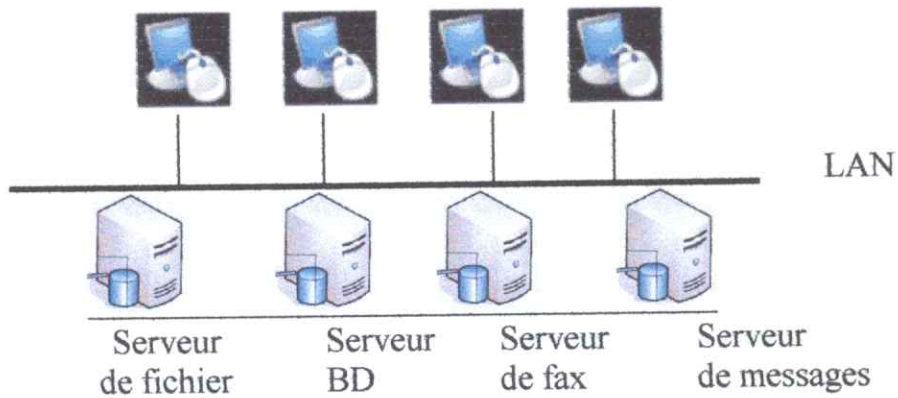


Figure 1.2: architecture client/serveur à serveurs multiples

Mais les environnements sont souvent plus complexes par le besoin d'intégrer de larges volumes d'informations stockés dans des bases de données distantes avec des coûts de production et de maintenance moindres.

Ces environnements intègrent souvent plusieurs serveurs répartis sur plusieurs réseaux interconnectés entre eux par des points et des passerelles pour que les utilisateurs répartissent leurs données sur différents serveurs, ce sont les systèmes distribués hétérogènes [KHO 98].

Ce type d'architecture nécessite de combiner constamment des informations provenant de différentes sources d'informations.

I.2.2. Les systèmes distribués :

L'intégration de sources de données réparties est une méthode qui vise à offrir aux utilisateurs l'accès à des multiples sources de données à travers des requêtes sur un schéma global. Les BD distribués (réparties) constituent un des applications les plus courantes des systèmes distribués.

Une BD distribuée dans le sens le plus large est considéré comme un ensemble de plusieurs BD mises en interrelations à travers un réseau.

Le système de gestion de BD distribuées « SGBDD » est l'application logicielle qui permet la manipulation des données.

- **Avantages de la distribution :**

Plusieurs avantages sont issus de l'approche distribuée dans la conception de systèmes d'information parmi ces avantages certains concernent les aspects humains, d'autres sont d'ordre organisationnel ou économique [SHE 90].

1. **Minimiser les coûts de communication :** un choix convenable de la distribution des données peut réduire les coûts de communication.
2. **Sécurité des opérations :** Avec la distribution de copies de données identiques sur plusieurs sites sans redondance, l'arrêt d'un système ne provoque pas l'arrêt de l'ensemble
3. **Flexibilité :** En ajoutant plusieurs sites au réseau, un système distribué peut étendre la base et le traitement pour de nouvelles applications. Ceci peut se produire lors de la fusion de deux sociétés qui devront, sans remettre en cause les systèmes de bases de données qu'ils avaient avant, avoir une vue globale intégrée sur ces deux systèmes.
4. **Performance :** le traitement peut être réparti sur plusieurs sites en assurant le maximum de parallélisme.

5. **Autonomie** : la répartition permet aux différents services d'information d'une même entreprise de conserver leur individualité, leurs méthodes de travail et leurs manières de voir les choses.
6. **Augmentation du rendement des capitaux investis** : les systèmes d'information est particulièrement ceux qui sont fortement automatisés sont indispensables au bon fonctionnement des organisations.

Les mutations technologiques de ces dernières décennies ont bouleversé profondément la vie des organisations. Progressivement, l'idée de coopération dans les systèmes d'information distribués est devenue une nécessité afin de répondre à des nouvelles exigences organisationnelles.

I.3. La coopération des SI :

A l'aube de ce nouveau siècle, les entreprises adoptent de nouveaux types d'organisation [ESP 95] leur permettant d'accroître leur flexibilité et leur réactivité , ces nouveaux types d'organisation présentent de nombreuses similitudes en mettant notamment l'accent sur deux points fondamentaux :

- la distribution de responsabilités et de capacités de décisionnelles de l'entreprise
- le développement de politiques de coopération entre ses unités.

Les possibilités offertes par les nouvelles techniques de traitement de l'information et de connaissances. Mais aussi et surtout le développement d'intranet et d'intranet, apportent de nouveaux challenges, notamment celui de développer de nouveaux SI permettant d'une part d'accéder a des données et connaissances multimédias de sources d'information localisées partout de par le monde et d'autre part de supporter la coordination d'activités organisationnelles humaines et /on artificielles associées aux nouveaux types d'organisation.

Progressivement, l'idée de coopération dans les SI est devenue une nécessité afin de répondre à ces nouvelles exigences organisationnelles. De nouvelles applications ont immergé nécessitant des accès coordonnés à des systèmes d'information distants et hétérogènes.

Au niveau informatique. Des solutions techniques sont basées sur la notion d'interopérabilité des systèmes qui reposent sur des concepts facilitant la transparence de l'échange de données et des services inter systèmes.

Cette nouvelle tendance de SI a donné naissance à une nouvelle génération de SI appelés *systèmes d'information coopératifs*.

Les systèmes d'information coopératifs sont généralement caractérisés par la grande variété et le grand nombre de sources d'information. Ces sources d'information sont hétérogènes et distribuées sur un réseau de communication [PAS 05].

I.3.1. Notion de coopération

De nombreux travaux sur la coopération ont montré l'intérêt d'une telle approche comme moyen d'optimisation des performances (collectives ou individuelles), basé des fonctions de satisfaction de coûts par exemple. La coopération est aussi un moyen pour des agents d'accomplir des tâches impossibles pour un individu isolé [FER 95] et donc obtenir des fonctionnalités de plus haut niveau, comme s'adapter ou apprendre [GLI 97].

La coopération entre acteurs ou agents d'un réseau résulte de leur capacité à collaborer, co-décider et coordonner leurs activités [D.MI 94]. Mais la codécision, mettant en jeu les mécanismes de négociation, peut être considérée comme une activité coopérative particulière qu'il est possible de décrire par des activités de communication, de coordination et de collaboration.

La figure 1.3 représente le mécanisme de négociation :

- une solution est proposée par un acteur (1) ;
- les autres acteurs approuvent cette solution ou proposent des modifications (2).
- Les modifications peuvent être refusées ou prises en compte pour la proposition de nouvelles solutions. Et ainsi de suite jusqu'à l'obtention d'une solution partagée par tous (3).

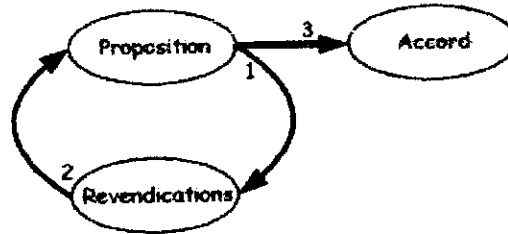


Figure 1.3 : Mécanisme de négociation

La coopération peut être définie de la façon suivante en s'inspirant de [SCH 97] :

Définition: *La coopération est une aptitude à la communication, la coordination et la Collaboration d'un ensemble d'acteurs pour la réalisation d'un objectif commun.*

- *La communication est indispensable* : deux acteurs ne peuvent coopérer sans échanger des informations, soit directement, soit indirectement par modification de leur environnement commun (par exemple l'ensemble des usages qui régissent le fonctionnement d'un bureau est mis au point progressivement dans le but de faciliter le travail de chaque acteur, sans communications directes préalables entre ceux-ci).
- *La coordination*, quant à elle, est nécessaire dans tous les types d'organisations, dans le cas du travail à la chaîne, par exemple, les actions de chaque opérateur sont fortement liées aux actions de l'opérateur qui le précède. La réalisation de l'objectif commun n'est possible que par une action coordonnée de l'ensemble des acteurs. De même, dans le cas des équipes sportives, la coopération prend souvent la forme de coordinations de mouvements tactiques ou techniques individuels.
- *La collaboration*, l'activité commune d'un ensemble d'acteurs, est une troisième caractéristique de la coopération. Elle est l'expression d'une activité synchrone. Par exemple, le tri postal est une activité commune d'acteurs travaillant dans un même espace de temps pour la réalisation de l'objectif commun qui est le tri de l'ensemble du courrier.

Au niveau de l'individu, la coopération est modulée par la confiance qu'il porte envers les autres acteurs. Cette confiance peut revêtir trois formes : elle peut être fonction de caractéristiques propres à la personne telle que l'appartenance à un groupe, à une famille, à

une ethnie ; elle peut également être liée aux précédents échanges, à la réputation de l'autre acteur (il s'agit alors de la confiance relationnelle) ; elle peut être liée à une structure formelle garantissant les activités de l'autre acteur (cette confiance est nommée *institutionnelle*) [ADA 00].

I.3.2. Méthodes de coopération

[FER 95] a cité six méthodes de coopération: le regroupement et la multiplication, la communication, la spécialisation, la collaboration par partage des tâches et de ressources, la coordination d'actions, la résolution de conflit par arbitrage et négociation.

Le terme agent est utilisé dans la suite pour représenter les éléments qui participent à la coopération (individus, ressources matérielles, ressources logicielles,.....)

1. **le regroupement et la multiplication** : cette méthode consiste tout simplement pour les agents à se rapprocher physiquement. C'est à dire à constitué soit un bloc plus au moins homogènes dans l'espace, soit un réseau de communication permettant à plusieurs agents de ce comporter comme s'ils étaient physiquement les uns à coté des autres.
2. **la communication** : le système de communication qui lie un ensemble d'agents sorte de système nerveux qui met en contact des individus parfois séparer. La communication en effet agrandit les capacités perceptives des agents en leur permettant de bénéficier des informations et du savoir faire des autres agents. Les communications sont indispensables à la coopération et il est difficile de concevoir un système d'agents coopérants s'il n'existe pas un système permettant aux agents d'échanger des informations ou de véhiculer des requêtes, la communication constitue l'un des moyens fondamentaux pour assurer la répartition des taches et la coordination des actions.
3. **la spécialisation** : la spécialisation est le processus par lequel des agents deviennent de plus en plus adaptés à leurs taches. il est souvent difficile de faire des agents qui soient spécialises dans toutes les taches. la réalisation performante d'une tache suppose souvent des caractéristiques structurelles et comportementales qui ne peuvent permettre d'effectuer d'autres taches avec efficacité. Des agents initialement totipotents peuvent ne spécialiser progressivement dans l'accomplissement de leurs

taches, ce qui revient à leur associer des rôles temporaires par une sorte d'adaptation individuelle. Cette spécialisation peut être bénéfique à la collectivité en augmentant la capacité du groupe à résoudre plus rapidement un problème semblable.

4. **la collaboration par partage de tâches et de ressources** : la collaboration consiste à travailler à plusieurs sur un projet, une tâche commune. Nous appellerons collaboration l'ensemble des techniques permettant à des agents de (se) répartir des tâches, des informations et des ressources de manière à réaliser une œuvre commune. Résoudre un problème de collaboration consiste donc à répondre à la question « qui fait quoi » par rapport à un travail donné. il existe de nombreuses manières de répartir des tâches et des moyens.
5. **la coordination d'actions** : gérer un ensemble d'agents suppose l'exécution d'un certain nombre de tâches supplémentaires 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. Lorsqu'il s'agit de système monolithique, ces tâches supplémentaires font partie du système organisationnel, mais dans le cas d'un SMA, c'est-à-dire d'un travail nécessairement distribué, on parle de tâches de coordination celles-ci sont indispensables dès lors que l'on se trouve en présence d'un ensemble d'agents autonomes qui poursuivent leurs propres buts, la réalisation de tâches productives entraînant avec elles tout un cortège de tâches de coordination sans lesquelles les premières ne peuvent être accomplies
6. **la résolution de conflit par arbitrage et négociation** : l'arbitrage et la négociation sont deux des moyens utilisés par les systèmes multi agents pour résoudre les conflits et empêcher que des désaccords entre individus ne se transforment en luttes ouvertes et que le système dans son entier ne dégrade ses performances. L'arbitrage conduit à la définition de règles de comportement qui agissent comme des contraintes sur l'ensemble des agents mais dont le résultat global a pour effet de limiter les conflits et de préserver à la fois les individus mais surtout les sociétés d'agents. lorsque des agents entrent en conflit d'objectif ou de ressource on préfère souvent ne pas recourir à un arbitre. Mais laisser plutôt les agents résoudre eux même le conflit par la recherche d'un bilatéral au cours d'un processus de négociation.

I. Niveau de coopération :

La coopération peut être située à différents niveaux lors de la réalisation de systèmes coopératifs [PAS 05] :

- **Coopération homme/machine** : est issue de l'interaction Homme / Machine, Il s'agit en quelque sorte d'une interaction Homme / Machine enrichie. Pour les concepteurs de systèmes, le problème consiste à doter la machine de capacités supplémentaires afin de guider l'utilisateur dans son processus de résolution de problèmes. L'idée se résume ici à concevoir des interfaces adaptatives intelligentes.
- **Système supportant des tâches coopératives** : le problème consiste ici à développer des systèmes capables de supporter des tâches collectives impliquant une coopération entre différents acteurs.
- **Système coopératif** : Le système en lui-même possède des fonctionnalités capables de supporter la coopération.

Nous nous intéressons dans ce qui suit aux différents outils pour systèmes coopératifs issus de différents courants de recherche :

1. TCAO (Travail Coopératif Assisté par Ordinateur).
2. Groupware.
3. Les outils de Workflow.
4. Système d'information coopératif à base d'agents.

1.3.4. Les outils pour systèmes coopératifs :

1) Le Travail Coopératif Assisté par Ordinateur (TCAO)

La notion d'organisation a toujours été dépendante de la notion de tâche ou de celle d'échange d'informations. Or, cette dernière décennie a connu l'émergence et l'utilisation généralisée de la messagerie électronique et surtout la mise en place de plus en plus systématique d'outils d'aide au travail coopératif.

Ces outils sont soit de type workflow [MAR 95], c'est-à-dire destinés à la gestion de flux de données (le terme GED pour Gestion Electronique des Documents est également utilisé), soit de type Groupware, plus orienté vers le travail de groupe sur un même ensemble de documents [SCH 97]. Ces outils, issus du domaine du Travail Coopératif Assisté par Ordinateur¹ (TCAO) [WIL 94] contribuent à la restructuration des organisations traditionnellement centralisées, au profit d'organisations distribuées dont les acteurs sont physiquement éloignés (notion d'entreprise distribuée ou virtuelle).

Les technologies de l'information et de la communication accompagnent le travail de groupe pour répondre aux trois problèmes induits par la division du travail à savoir la communication, la coopération et la coordination. Elles permettent ainsi d'être moins dépendant des contraintes de lieu et de temps induites par la mondialisation et même d'accroître la productivité.

Le travail coopératif assisté par ordinateur TCAO (ou en anglais CSCW [computer supported cooperative work]) est donc la réponse technique à une évolution managériale profonde.

- **Le domaine du TCAO :**

Le domaine du TCAO a pour thème d'étude les systèmes interactifs multi utilisateurs permettant à plusieurs utilisateurs de travailler ensembles.

Le domaine du TCAO doit adapter et étendre les méthodes et outils développés en IHM car ceux-ci ont été forgés par rapport à une activité mono utilisateur et en fonction d'un seul utilisateur. Ce constat est particulièrement saillant en ce qui concerne les facteurs humains (sciences humaines).

Tandis que le domaine d'IHM s'est longtemps appuyé sur des théories de psychologie cognitive et comportementale, le domaine du TCAO dès ses débuts s'est tourné vers les sciences sociales et en particulier l'ethnographie pour analyser les aspects sociaux au sein d'un groupe en situation de travail [WIL 94].

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

2) Groupware :

Les systèmes coopératifs renvoient généralement au concept de groupware. Ces travaux sont issus du courant de recherche Computer Supported Collaborative Work (CSCW). La notion de « groupware » en français collecticiel est apparue durant les années 80.

Le Groupware désigne l'ensemble des technologies et méthodes de travail associées qui, par l'intermédiaire de la communication électronique, permettent le partage de l'information sur un support numérique à un groupe engagé dans un travail collaboratif et/ou coopératif [PAS 05].

D'un point de vue technologique le groupware se situe aux frontières de l'informatique et des télécommunications. En effet des technologies issues de l'informatique ainsi que de la télécommunication sont utilisées.

D'un point de vue système d'information le groupware se situe aux frontières de la bureautique et de l'informatique transactionnelle. En effet, d'un point de vue utilisateur les outils de groupware font partie de la gamme des outils de bureautique.

Le Groupware accompagne l'évolution de l'organisation des entreprises, il répond à la demande de qualité des services rendus par l'entreprise (rapidité, efficacité, ...), et il permet

un travail collectif au sein de groupes plus ou moins larges, distribués géographiquement et organisationnellement.

Ainsi le terme **Groupware** renvoie à des applications diverses et variées concourant à un même but : permettre à des utilisateurs géographiquement éloignés de travailler en équipe. Le travail en équipe peut se concrétiser par le partage d'information ou bien la création et l'échange de données informatisées.

Il s'agit pour la plupart du temps d'outils de messagerie (instantanée ou non), ainsi que d'applications diverses telles que :

- agenda partagé,
- espace de documents partagés plus ou moins organisés,
- outils d'échange d'information (forums électroniques),
- outils de gestion de contacts,
- outils de workflow,
- conférence électronique (vidéoconférence, chat,...).

Les principales fonctionnalités des outils de groupware résident dans :

- La communication interpersonnelle grâce à la messagerie électronique, aux forums de discussion,
- La coordination (temps, espace, tâches) grâce aux agendas électroniques partagés, ainsi que la gestion de projet,
- La collaboration grâce aux réunions électroniques, visioconférences, ainsi que l'édition conjointe de documents,
- La mémoire du groupe ainsi que l'accès à l'information grâce aux SGBD de documents,
- L'automatisation des processus administratifs grâce aux outils de workflow.
- Systèmes d'Information Coopératifs.

❖ Défis et problème du groupware

Le fait qu'il existe désormais de nombreux produits qui se réclament du groupware ne doit pas faire oublier qu'il reste des problèmes à résoudre et des leçons à tirer avant qu'il soient définitivement adoptés et correctement utilisés.

On notera que le problème majeur est celui de la complexité des rapports humains, qui se retrouve transposée dans le groupware [KHO 98].

- **Uniformité des produits**

Dans le cas de groupware et du workflow, les utilisateurs doivent employer le même produit et se montrer prêts à coopérer en respectant les règles et les procédures d'usage de ce produit.

Un système d'agenda partagé sera inutile si les travailleurs n'en font pas d'usage.

- **Trop de structure ou trop de travail**

Le problème de la structure est un problème sérieux qui mérite d'être étudié, si une fois adopté, un outil nécessite trop de structure et de logistique pour qu'il soit mis en œuvre.

- **Remettre en cause et aplanir les infrastructures organisationnelles**

Il est nécessaire de noter qu'en termes de communication et d'accessibilité, la messagerie électronique a eu des effets très importants de réduction des niveaux de management, malgré les résistances des managers intermédiaires.

- **Une distribution injuste des tâches**

Si la solution groupware adoptée implique que les travailleurs travaillent plus que d'autres parce que certaines tâches sont plus automatisées que d'autres, la solution retenue court à l'échec.

- **Un manque évident de gestion des exceptions**

Le groupware n'essaie pas d'automatiser des usines, il essaie d'améliorer la productivité des individus et les complexités en jeu sont monstrueuses.

- **Résistance au changement**

Ce problème peut mettre en danger la mise en œuvre d'une solution de groupware. En fait dans bien des cas, l'investissement dans une solution de groupware se justifie en termes d'amélioration de la productivité, toujours difficile à prouver.

3) Workflow

Il existe une grande variété de définitions pour le workflow. Nous n'en citerons que deux, une définissant ce que sont les logiciels de WorkFlow, et l'autre définissant ce que sont les procédures, pièces centrales du WorkFlow :

- Logiciel de WorkFlow : "Ensemble de logiciels proactifs qui permettent de gérer les procédures de travail, de coordonner les charges et les ressources et de superviser le déroulement des tâches"
- Procédure : "Tout ensemble de tâches exécutées en parallèle ou en série par au moins deux membres d'un groupe pour atteindre un but commun" [PAS 05].

Cependant dans le domaine du Workflow, il existe un certain nombre de concepts clés à définir.

❖ Types de workflow

Dans les applications de WorkFlow, on distingue classiquement quatre catégories [KHO 98] :

- le workflow de production, qui correspond à la gestion des processus de base de l'entreprise. Les procédures supportent peu de changements dans le temps, et les transactions sont répétitives. On peut y trouver par exemple la production de contrats d'assurance, la gestion de litiges, la gestion de réclamations clients, etc.
- le workflow administratif, qui correspond à tout ce qui est routage de formulaires, basé en général sur une infrastructure de messagerie.
- le workflow ad hoc pour la gestion des procédures non déterminées, ou mouvantes.
- le workflow coopératif, gérant des procédures évoluant assez fréquemment, et liées à un groupe de travail restreint dans l'entreprise.

❖ Types d'invocation

Dans un système de workflow, il existe traditionnellement deux façons d'accéder à l'information :

- Soit en allant la chercher dans une file d'attente partagée

Il s'agit d'une invocation de type base de données, une file d'attente étant gérée comme une table à laquelle accèdent les différents acteurs d'une procédure.

L'avantage de cette méthode d'invocation est la possibilité de faire partager un ensemble de tâches par un groupe d'utilisateurs.

En effet, lorsqu'une tâche est affectée à un profil donné, on ne va pas forcément l'affecter à une personne donnée. On mettra l'ensemble des tâches à disposition de toutes les personnes correspondant au profil, et on aura ainsi une régulation de la charge qui pourra se faire au mieux. La contrepartie est qu'il faut aller chercher la liste des tâches pour l'obtenir, en utilisant une infrastructure propre à l'outil de workflow. Il faut mettre en place également des stratégies de rafraîchissement de la liste des tâches, car on ne peut être en permanence en train de scruter la file d'attente.

- Soit en la recevant dans sa boîte de réception personnelle.

Il s'agit d'une invocation de type messagerie, puisque le meilleur moyen d'envoyer une tâche à exécuter à une personne est de la lui notifier à travers sa messagerie, et donc à travers sa boîte de réception. On utilise ici une infrastructure existante, et un mode de communication standard par rapport aux habitudes d'un utilisateur. Les logiciels de WorkFlow fonctionnant sur ce principe sont donc plus légers, et l'utilisateur final à l'impression de traiter ses tâches tout comme il traite son courrier électronique. Il ne va pas chercher spécifiquement les tâches qu'il a à effectuer, puisqu'elles arrivent directement via sa messagerie sur son poste de travail.

Classiquement la plupart des logiciels de workflow se répartiront ainsi entre les deux catégories. En terme de performances, les produits s'appuyant sur l'infrastructure de messagerie sont beaucoup plus légers, donc plus faciles à mettre en oeuvre, et générant en particulier des charges réseau moins importantes, puisqu'ils s'intègrent dans une infrastructure existante. Certains n'utilisent même pas du tout de base de données, et se contentent de fournir des pilotes leur permettant de se connecter aux bases existantes dans l'entreprise [PAS 05].

❖ La nature des tâches

On trouve essentiellement trois types de tâches dans un système de WorkFlow. Certains systèmes n'en supporteront qu'un, d'autres en supporteront d'emblée deux ou trois. Les types de tâches sont les suivantes :

- les tâches qui sont en fait des formulaires de données, généralement définis à partir du logiciel de workflow lui-même, à compléter au fur et à mesure de l'avancement de la procédure. Ce sont les tâches que l'on trouve dans les workflow administratifs.
- les tâches qui sont des services du système d'informations, tels que la saisie de transactions gros systèmes, ou l'appel à un exécutable spécifique, etc... Le workflow de production est entièrement basé sur ce genre de tâches, puisque son travail consiste à coordonner l'ensemble des actions possibles au sein du système d'information.
- Les tâches qui correspondent à un routage de fichiers bureautiques. On retrouve ces tâches essentiellement dans les workflow administratifs ou dans les workflow ad hoc, comme peuvent le proposer les différents traitements de texte du marché en se servant des messageries comme infrastructure.

4) Système coopératif à base d'agent

L'approche multi agent est utilisée pour la mise en oeuvre d'un système qui permet d'intégrer plusieurs applications existantes. Elle facilite leurs coopérations ainsi que le partage des informations communes. Cette approche à savoir les SMA, est décrite de manière détaillée dans le chapitre 2.

L'avalanche des nouveautés technologiques tels que les systèmes client/serveur pour l'aspect architecture, l'approche objet pour l'aspect méthodes d'analyse et modélisation, les Groupware et Workflow pour l'aspect échange et coopération à engendrer des difficultés d'élaboration des SI fiables et modernes au service de la stratégie de l'entreprise.

Pour cela il faut que les architectes de Systèmes d'Information doivent concevoir des systèmes offrant les mêmes fonctionnalités que les systèmes d'information classiques mais devant par ailleurs supporter le travail collaboratif et coopératif au sens large, cette nouvelle génération de SI est appelée *systèmes d'information coopératifs*.



II. SIC : Systèmes d'Information Coopératifs

Les Systèmes d'Information Coopératifs (SIC) assistent le travail collectif entre différents types d'agents (individus, services, entreprises, etc.). De tels SI possèdent des spécificités intrinsèques liées au caractère potentiellement hétérogène des sources d'information, des processus et des savoirs mis en relation et à l'indépendance que les agents souhaitent (ou doivent) conserver. Ce type de système favorise d'un côté, le nomadisme et la mobilité des agents avec les mêmes fonctionnalités et niveaux de sécurité qu'en intra entreprise, et d'un autre côté, l'intégration des entreprises (par les données, par les applications, par les processus ou encore par les connaissances) dans le cadre de l'entreprise étendue (sous-traitants, clients, fournisseurs, etc.). C'est un domaine en devenir qui emprunte à des domaines variés : système d'information, intelligence artificielle distribuée, bases de données réparties, sciences de gestion, sociologie, sciences de l'éducation, etc.

De tels systèmes doivent être capable d'exécuter principalement les tâches suivantes :

- *la découverte des ressources* : trouver la bonne source pour l'interroger.
- *la recherche d'informations* : identifier les informations non structurées et semi structurées.
- *le filtrage des informations* : analyser les données et éliminer celles qui sont inutiles, un système de filtrage de l'information peut assortir l'information qui est appropriée ou importante des énormes quantités de l'information et filtrer l'information non pertinente et sans importance.
- *la fusion des informations* : regrouper les informations d'une manière significative, le volume d'information disponible empêche de trouver réellement l'information pour répondre à des questions spécifiques.

II.1 Propriétés des Systèmes d'Information Coopératifs :

Les propriétés des SIC sont au nombre de trois, ont été évoquées dans différents articles [SCI 94], [LUI 95]..., ces propriétés sont : distribution, autonomie et hétérogénéité.

II.1.1. Distribution :

La question de la distribution physique des sources de données est orthogonale à l'autonomie et à l'hétérogénéité des systèmes. Depuis déjà quelques années, il est naturel de penser que les données et les traitements ne soient pas physiquement sur un même lieu ou machine mais, au contraire, repartis sur un réseau. Des techniques comme CORBA de l'OMG (Object Management Group), permettent cette distribution

II.1.2. Autonomie :

L'autonomie fait référence au contrôle indépendant et sépare des systèmes composants. Ainsi l'intégration ou l'abandon d'un composant ne doit pas perturber le fonctionnement global du SIC. En général, l'autonomie apparaît sous différentes formes. [SHE 90] a identifié trois niveaux d'autonomie :

- l'autonomie de conception : indique que chaque système composant dans le SIC est libre de choisir sa propre conception. Cependant, il doit respecter certaines contraintes comme le modèle de données, le langage de requêtes, la représentation de données et les contraintes d'intégrité et d'implémentation.
- L'autonomie de communication : se réfère à la capacité d'un système composant de décider de la façon de communiquer avec les autres systèmes. Un système serait capable de décider quand et comment pourrait répondre aux requêtes provenant des autres systèmes composants.
- L'autonomie d'exécution : permet à un système d'exécuter ces opérations locales sans interférence avec les opérations soumises par d'autres systèmes composants. Ceci implique q'un système ne peut pas imposer un ordre d'exécution à un autre système ayant ce type d'autonomie.

II.1.3. hétérogénéité :

L'hétérogénéité peut se manifester sous plusieurs formes. Elle peut être provoquée par les différences technologiques au niveau des plates-formes matérielles et/ou logicielles (systèmes d'exploitation, réseau de communication, etc.). On peut distinguer également l'hétérogénéité induite par les différences des systèmes d'information composants. Ces différences se situent en générale au niveau des modèles de données pour l'expression des structures et des contraintes et au niveau des langages de requêtes.

Les différentes formes d'hétérogénéités engendrent plusieurs types de conflits lors de la coopération des systèmes d'information.

Dans ce qui suit, nous présentons la classification proposée par Busse [BUS 99], dans cette classification nous distinguons trois types d'hétérogénéités (figure 1.4) : l'hétérogénéité syntaxique, du modèle de données et logique que nous détaillons ci-après.

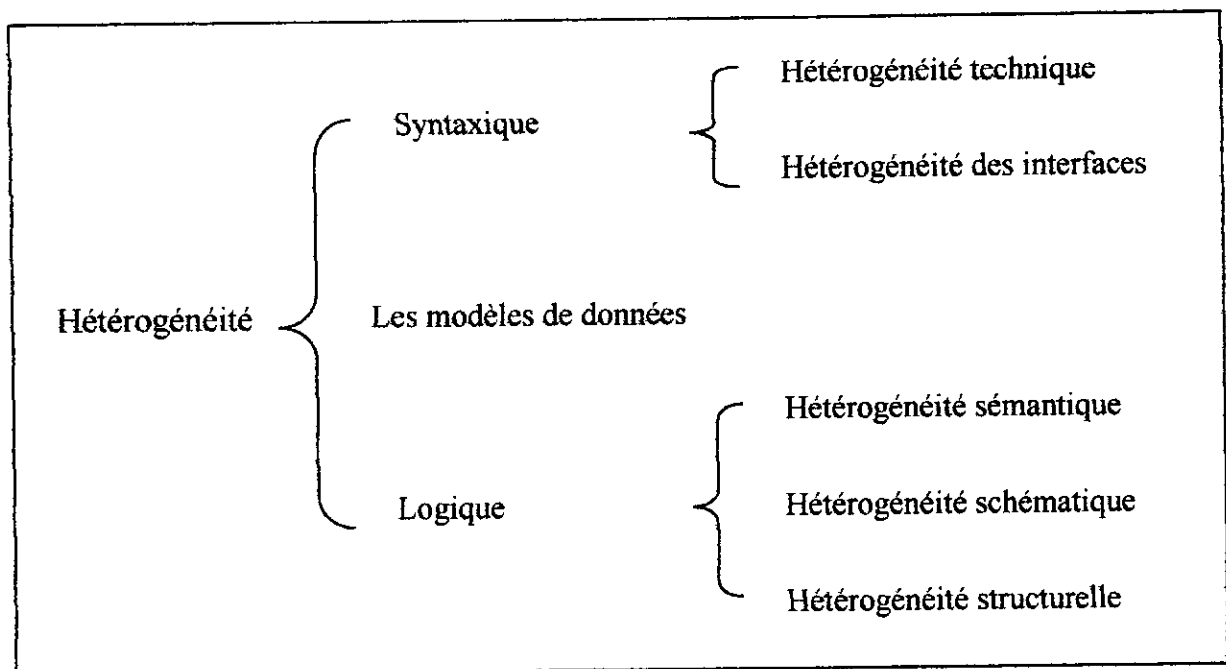


Figure 1.4: classification de l'hétérogénéité

a) hétérogénéité syntaxique :

L'hétérogénéité syntaxique peut être technique ou liées aux interfaces :

- *L'hétérogénéité technique* : couvre différents aspects techniques comme les plates formes d'exécution et les systèmes d'exploitation, les méthodes d'accès (protocoles, méthodes de connexion, etc.)
- *L'hétérogénéité des interfaces* : existe si différents composants sont sollicités au travers des langages différents. Il ne s'agit pas dans ce cas des choix technique mais plutôt des choix des méthodes d'accès :
 - Hétérogénéité des langages : différents langages de requêtes avec des restrictions potentielles
 - Restrictions de requêtes : requêtes autorisées ou non, expression d'un nombre restreint de condition, jointures limitées, etc.

b) Hétérogénéité des modèles de données :

L'hétérogénéité des modèles de données exprime le fait que différents modèles de données associent des contenus sémantiques différents à leurs concepts. Par exemple, un modèle relationnel n'a pas de concept d'héritage contrairement à un modèle orienté objet. Il y a alors des concepts qui ne sont pas couverts par l'ensemble des modèles de données des différents composants. Bien que l'hétérogénéité des modèles des données soit un cas particulier d'hétérogénéité sémantique et structurelle, elle est souvent mise à part dans un nombre important de systèmes.

c) hétérogénéité logique :

L'hétérogénéité logique peut être sémantique, schématique ou structurelle :

▪ *L'hétérogénéité sémantique :*

Traduit le fait que des données dans différents systèmes sont sujettes à des interprétations différentes, même si les schémas qui leur correspondent sont identiques. A l'origine ce problème provient de l'écart séparant le monde réel du monde conceptuel (le

second est une représentation du premier) cela signifie que si l'on considère un objet dans le monde réel on peut le représenter de diverses manières dans le monde conceptuel. Deux types de conflits sémantiques ont été mis en évidence [SHE 92] [BEN 99] :

- Hétérogénéité de nommage : consiste en des relations de synonymie (un même concept est décrit par des noms différents) ou d'homonymie (un même nom peut être utilisé pour dénoter deux concepts différents) entre les valeurs des attributs. Par exemple, un attribut « route » peut avoir comme domaine de valeurs l'ensemble {nationale, départementale} dans un système et l'ensemble {Nat, Dép.} dans un autre.
- Hétérogénéité d'échelle ou d'unité : concerne l'utilisation d'unités ou d'échelles différentes pour mesurer les valeurs. Par exemple, le prix d'un livre peut être exprimé en Dollar dans un système, et en Euro dans un autre.

▪ ***L'hétérogénéité schématique :***

Concerne l'encodage des concepts par différents éléments du modèle de données. Par exemple, dans le modèle relationnel, trois types de conflits ont été recensés [KRI 91] :

Relation	↔	nom de l'attribut,
Nom de l'attribut	↔	valeur de l'attribut,
Relation	↔	valeur de l'attribut.

▪ ***L'hétérogénéité structurelle :***

Elle apparaît lorsque des éléments identiques ayant le même sens (contenu sémantique identique) et modélisés avec le même modèle de données sont schématiquement homogènes mais structurés de différentes façons. Par exemple un concepteur peut définir deux concepts pour distinguer les salariés des personnes non salariées alors qu'un autre concepteur peut utiliser un seul concept pour représenter toutes les personnes.

II.2 Techniques de mise en œuvre de coopération :

Dans cette partie nous présentons les techniques de coopération selon deux approches :

1. **les systèmes d'information fédérés** : la coopération est assurée par l'interopérabilité des sources d'information autonomes distribuées et hétérogènes.
2. **les systèmes de coopération à base de multi agents (SMA)** : dans le cadre des systèmes multi agents la coopération présuppose des agents qui ont des buts et qui peuvent agir entre eux. Plus généralement un agent est coopératif s'il partage des buts avec d'autres agents dans son environnement et agit pour l'accomplissement de ces buts communs [MIC 97].

II.2.1. Les systèmes d'Information Fédérés (SIF) :

Les systèmes d'information peuvent être classifiés selon les critères de la distribution et de l'hétérogénéité [BUS 99] comme suit :

- Un système unique (monolithique, centralisé) s'exécute comme une application monolithique, sur une seule machine. Un système d'information monolithique peut être soit structuré soit semi structuré.
- Un système d'information distribué est un système pour lequel les données sont physiquement distribuées sur plusieurs sites interconnectés.
- Un système d'information hétérogène est une collection de systèmes qui diffèrent selon des aspects logiques ou syntaxiques (plate forme.....).

Si nous ajoutons la dimension de l'autonomie nous abordons la définition d'un système d'information fédéré SIF, vu comme un ensemble de composant (ou systèmes) distincts et indépendants : les participants de la fédération.

Dès la fin des années 80, l'objectif des premières propositions relatives à la coopération de systèmes d'information a été de traiter les différences techniques, structurelles et syntaxiques entre les systèmes. Les solutions représentatives de cette période sont *les*

systemes de bases de données fédérées, qui correspondent à une intégration partielle des bases de données existantes.

Un Système de Base de Données Fédérées (SBDF) est une collection de systèmes de base de données coopératifs, autonomes et éventuellement hétérogènes.

Le système fédéré est constitué d'un ensemble de systèmes de bases de données pouvant être de différents types, ce qui peut donner une structure très complexe pour la fédération [SHE 90].

L'approche fédérée est orientée vers l'intégration des structures de données, soit sous forme d'un schéma fédéré (couplage fort), soit sous forme de vues fédérant plusieurs SI (couplage faible).

1) Objectifs et problématique :

L'approche fédérée consiste à étendre le schéma classique à trois niveaux (niveau interne, niveau conceptuel, niveau externe) à un schéma sur cinq niveaux permettant un accès aux données de SI hétérogènes et distribués. Sheth dans [SHE 90] spécifie ces cinq niveaux de représentation (figure 1.5) :

- Le premier niveau est la représentation du schéma local de chaque système dans son modèle local.
- Le second niveau est l'expression de ce schéma local dans le modèle commun retenu par la fédération.
- Le troisième niveau est un ensemble de schémas d'export qui présente les informations sous des formats ou des conceptualisation différente des informations.
- Le quatrième niveau correspond aux schémas fédérés obtenus par intégration de schémas d'exports et correspondant à un domaine d'application.
- Le cinquième niveau est constitué de vues dérivées d'un schéma pour s'adapter à des catégories d'utilisateurs.

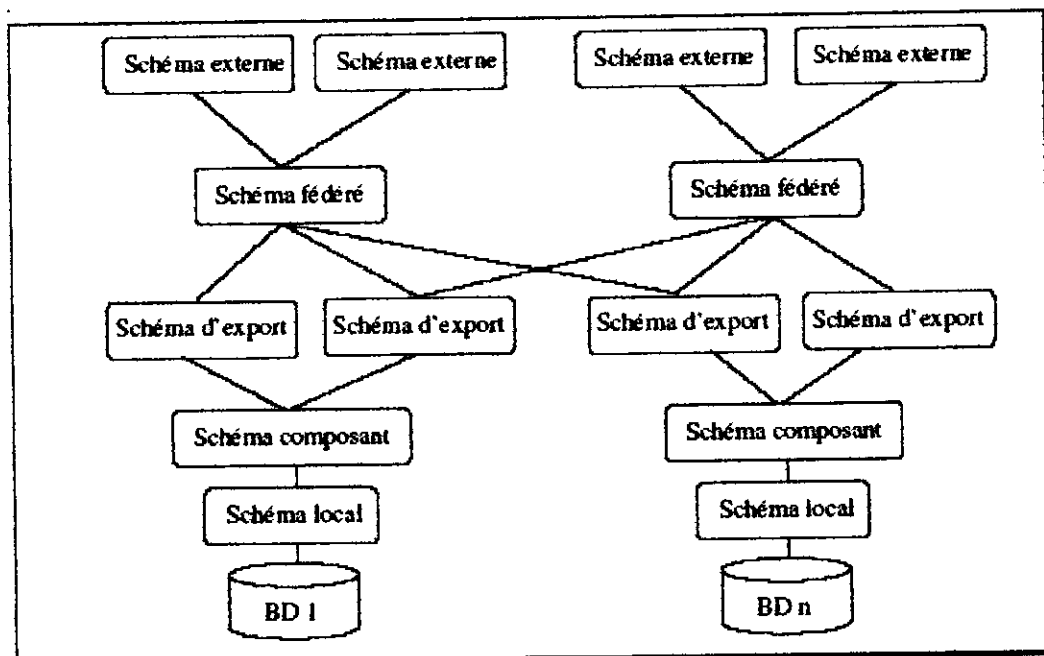


Figure 1.5 : Architecture de fédération à cinq niveaux [SHE 90]

La problématique est centrée sur des problèmes d'intégration de schémas qui passent par la résolution des conflits de données. Les conflits structurels sont explicités dans les différents schémas d'exports mais les conflits sémantiques restent implicites et doivent être pris en compte lors de l'intégration. Deux types d'approches fédérées sont mis en évidence : la fédération fortement couplée qui construit un schéma fédéré et la fédération faiblement couplée qui utilise un langage multi bases pour interroger et combiner les informations contenues dans les schémas d'export.

2) Types de fédération :

Il existe deux types de fédération, fédération faiblement couplée et fédération fortement couplée.

▪ Fédération faiblement couplée :

Cette approche se caractérise par l'absence d'un schéma global. Elle repose sur l'utilisation d'un langage d'interrogation et d'un modèle de représentation commun

Chaque SI exporte ses informations sous la forme d'un schéma décrit dans le modèle commun (généralement un modèle orienté objet), le langage multi bases (extension de SQL)

qui permet soit d'interroger plusieurs systèmes à la fois soit de définir des vues sur plusieurs systèmes [LIT 86] permet une interrogation multi sites.

L'objectif de la fédération faiblement couplée est l'intégration des informations des différents systèmes au coup par coup en fonction des besoins.

Les systèmes à couplage faible se caractérisent par une grande autonomie et un faible degré d'intégration. Leur autonomie peut atteindre le point où la présence des autres membres peut être ignorée. Le choix d'une faible intégration peut engendrer des problèmes dans la mise en œuvre des applications portant sur le SGBD multiples, puisque ce type de systèmes est caractérisé par l'absence d'un schéma global unique, son contrôle et son maintien sont réservés aux utilisateurs de la fédération.

▪ **Fédération fortement couplée :**

L'objectif de la fédération fortement couplée est la construction d'un schéma fédéré qui intègre les différents schémas d'exports des SI. Les utilisateurs effectuent les mises à jour et pose des requêtes sur ce schéma fédéré. L'approche suit généralement l'architecture en cinq niveaux de Sheth. Le schéma fédéré résout tous les conflits syntaxiques et structurels : un seul modèle et une seule représentation des informations corrént des données distribuées et hétérogènes. Il résout également implicitement les conflits sémantiques : le schéma fédéré représente l'organisation des informations pour un domaine d'application particulier.

Un SGBD fédère fortement couplé offre à l'utilisateur de la fédération une transparence des conflits des données.

Ces différentes approches d'intégration de source de données présente chacune une alternative différentes pour le développement des SI coopératifs.

II.2.2. Les systèmes de coopération à base de système multi agents :

Les SMA constituent un des axes de l'Intelligence Artificielle Distribuée (IAD). A la différence de l'Intelligence Artificielle (IA) classique, qui modélise le comportement intelligent d'un seul agent (aspect individuel). L'IAD s'intéresse entre autre à des comportements intelligents qui sont le produit de l'activité coopérative de plusieurs agents (aspect collectif).

On définit généralement un SMA comme un modèle informatique composé d'entités de base les agents. Ces derniers possèdent une autonomie en terme de décisions et d'actions et sont organisés en société au sein d'un environnement dans lequel et avec lequel ils interagissent.

Ils doivent de ce point de vue être capable de percevoir de décider, d'agir et de communiquer [FER 95] [HER 01]. Les SMA possèdent ainsi l'avantage de faire intervenir des schémas d'interaction sophistiqués, les types courants d'interactions incluent :

- **La coopération** : travailler ensemble à la résolution d'un but commun.
- **La coordination** : organiser la résolution d'un problème de telle sorte que les interactions bénéfiques soient exploitées.
- **La négociation** : parvenir à un accord acceptable pour les parties concernées.

Nous reprenons de manière plus détaillée les systèmes multi agents dans le prochain chapitre.

II.3. Problématique de la coopération de SI :

La coopération de système est confrontée à de nombreux problèmes d'hétérogénéités, et doit tenir compte de l'aspect ouvert et dynamique des environnements modernes.

Les hypothèses liées à la problématique de la coopération des SI reposent sur des principes de transparence, d'autonomie de distribution et d'hétérogénéité [SHE 90].

- **Transparence vis-à-vis de l'utilisateur** : l'utilisateur pense n'utiliser qu'un seul système d'information.

- *L'autonomie des systèmes composants* : chacun des SI participant à la coopération sait résoudre seul, certaines requêtes posées par l'utilisateur, il est libre de participer ou non aux tâches globales.
- *Répartition des informations* : assuré à travers les succès des standards d'interopérabilité qui permettent une réelle interopérabilité opérationnelle [BUK 96].
- *L'hétérogénéité* : l'hétérogénéité concerne les modèles de représentation de l'information, des langages de définition et de manipulation de données, des domaines d'application et des applications elles-mêmes.

Le nombre et la variété de sources et de services de données augmentent considérablement chaque jour. Pendant que plus d'information deviennent disponible, il devient plus difficile d'accéder aux documents désirés.

Conclusion :

Nous pouvons affirmer que les Systèmes d'Information prennent de nos jours une place centrale dans les organisations. Dans ce chapitre nous avons présentés les propriétés des SIC et nous avons abordé les propriétés et les techniques de coopération que nous avons classés en deux catégories :

Les systèmes d'information fédérés et les systèmes coopératifs à base de systèmes multi agents.

Les SMA permettent de concevoir un SIC comme un ensemble d'entités spécialisées et coopérantes, de plus l'approche Agent prend plus facilement en compte le caractère ouvert, distribué et concurrent des systèmes modéliser, en même temps qu'elle en facilite la conception, en apportant d'avantage de modularité et des abstractions de haut niveau.

Chapitre II :

Les Systèmes Multi Agents (SMA)

L'approche multi agent est utilisée pour la mise en oeuvre d'un système qui permet d'intégrer plusieurs applications existantes. Elle facilite leurs coopérations ainsi que le partage des informations communes. Cette approche permet aussi de découper le problème en un ensemble de tâches qui seront désormais effectuées par des programmes agents autonomes.

Il reste, donc, à prouver que cette approche est bien appropriée au domaine des SI coopératifs et qu'elle répond aux exigences mentionnées auparavant. Pour cela nous allons consacrer le chapitre suivant pour les systèmes multi agents.

1. Historique

Actuellement les systèmes informatiques deviennent de plus en plus complexes et répartis sur différents sites. La nécessité de passer du comportement individuel au comportement collectif, a poussé la communauté de *l'Intelligence Artificielle* (IA) classique à se tourner vers des systèmes plus autonomes, plus coopérants afin de pouvoir résoudre des problèmes complexes.

Le terme « intelligence artificielle » a été utilisé pour désigner un projet de recherche consistant à concevoir une machine intelligente, c à d capable de réussir aussi bien qu'un être humain dans des tâches jugées complexes. Les activités simples ou complexes, telles que la résolution de problèmes, l'établissement d'un diagnostic, la coordination d'action ou la construction de systèmes sont le fruit d'une interaction entre entités relativement autonomes et indépendantes, appelées *agents*, qui travaillent au sein de communautés selon des modes parfois complexes de coopération, de conflit et de concurrence, pour survivre et se perpétuer [FER 95].

Un agent rationnel "système expert" (première génération de programmes informatiques évolués) était censé de résoudre tout seul les problèmes. L'accent a donc été mis progressivement à partir de la fin des années 70 sur une résolution distribuée de problèmes, par la coordination d'un certain nombre d'agents, ce que l'on a alors commencé à appeler des « *systèmes multi agents* ». On utilise également le terme quasiment équivalent "*Intelligence Artificielle Distribuée*" IAD pour bien montrer l'opposition à l'IA classique, autarcique et centralisée [BRI 01].

Les SMA ont un rôle essentiel à jouer en s'inscrivant comme les possibles successeurs des systèmes à objets, en ajoutant à la localité des comportements l'autonomie et la répartition des prises de décisions.

Le système multi agents était une solution adéquate vu qu'il pouvait répondre aux nouveaux besoins applicatifs tels que : l'hétérogénéité, l'autonomie, l'ouverture, l'évolutivité, la flexibilité...

2. Définition d'un agent :

Avant de présenter un système multi agents, définissons en premier lieu la notion d'agent. Dans la communauté scientifique plusieurs définitions d'agent sont proposées mais tournant autour du concept qu'un agent est une entité autonome localisée ou non qui agit sur son environnement et possédant un réseau d'acointances. Voici quelques-unes des définitions énoncées dans la littérature :

- [FER 95] : on appelle agent une entité physique ou virtuelle :
 - Qui est capable d'agir dans un environnement
 - Qui peut communiquer directement avec d'autres agents.
 - Qui est mue par un ensemble de tendance (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser).
 - Qui possède des ressources propres
 - Qui est capable de percevoir (mais de manière limitée) son environnement.
 - Qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune).

- Qui possède des compétences et offres des services.
- Qui peut éventuellement se reproduire.
- Dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

➤ Un agent est un système informatique capable d'agir de manière autonome dans un certain environnement afin de répondre à ses objectifs de conception [WOO 02].

• **Caractéristiques des Agents**

Plusieurs propriétés sont reliées aux agents mais un agent ne possède pas forcément toutes ces propriétés. Celles qui sont le plus énoncées dans la littérature sont :

- *L'autonomie* : C'est-à-dire un agent doit pouvoir agir et décider seul des actions qu'il doit effectuer suivant les informations provenant de son environnement et des autres agents.
- *La réactivité* : ceci suppose qu'un agent doit pouvoir réagir en conséquence des informations perçues.
- *La pro activité* : l'autonomie est directement reliée à la pro activité. Un agent est proactif si son comportement est piloté par des buts et s'il est capable de prendre des initiatives afin de les atteindre.
- *L'efficacité* : un agent doit être capable d'atteindre ses buts et de résoudre les problèmes auxquels il fait face. Des statistiques et des estimations sont souvent faites pour mesurer l'efficacité d'un agent.
- *La capacité d'apprentissage* : un agent doit s'adapter à des situations nouvelles en se basant sur son expérience et ainsi acquérir de nouvelles connaissances.
- *La robustesse* : bien qu'un agent doit avoir la capacité d'apprendre, il ne doit pas évoluer rapidement à la moindre alerte et ainsi perdre son intégrité en étant sensible à des faux signaux.
- *La capacité à communiquer* : un agent est considéré comme étant une entité sociale ; donc il doit pouvoir communiquer avec les autres agents. La communication est une forme d'interaction dans un système multi agent. Ce point sera détaillé ultérieurement.

3. Différentes catégories d'Agents

Nous présentons dans cette partie les différents modèles d'agents, afin de comprendre leurs caractéristiques et leurs modes de fonctionnement.

On peut classer les agents selon plusieurs critères mais par la suite on tiendra compte d'un seul mode de classification qui est la granularité et qui détermine le processus de décision de l'agent. La taxonomie des agents se fait selon :

- *La granularité* : agent réactif ou agent cognitif.
- *La mobilité* : agent mobile (ou nomade) et agent stationnaire.
- *La fonctionnalité* : agent logiciel, agent d'information, agent de commerce, agent assistant.

Nous distinguons deux grandes familles d'agents les agents réactifs et agents cognitifs.

3.1 Agents cognitifs

C'est le premier modèle d'agents qui a été proposé. Il est nommé aussi agent délibératif, il permet de planifier les actions d'un agent au sein de son environnement [FIK 71].

En effet, les agents cognitifs sont capables à eux seuls de réaliser des opérations relativement complexes.

Généralement, ils coopèrent les uns avec les autres pour atteindre un but commun (résolution d'un problème, une tâche complexe, etc.). Ils possèdent un ensemble de représentations explicites (sur l'environnement, sur les autres agents et sur eux-mêmes) décrits dans une base de connaissances sur laquelle ils peuvent raisonner. Ils réagissent en fonction de leurs connaissances, leurs buts, de leurs échanges d'informations avec les autres agents et de la perception de l'environnement (voir figure 1.6). Ils sont dotés de moyens et mécanismes de communication pour gérer les interactions avec d'autres agents (coopération, coordination et négociation).

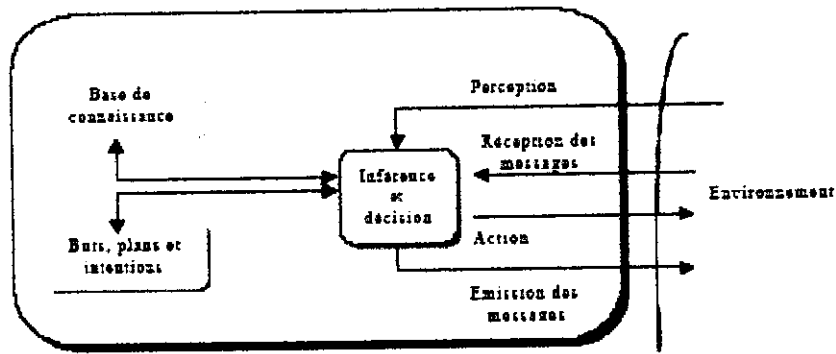


Figure 1.6 : Modèle d'un agent cognitif

Ce modèle d'agent est une métaphore du modèle humain et s'appuie sur la sociologie des organisations [LIU 02]. Nous pouvons aussi trouver son origine dans la volonté de faire coopérer des systèmes experts classiques dans le domaine d'IA.

L'agent cognitif, en terme de BDI pour Beliefs (croyances), Désires (désirs) et Intention (intentions), [BRA 87 ; BRA 88], postule que ses actions, tout comme celles d'un être humain, doivent être dictées par des représentations abstraites du monde, incluant la représentation de ses propres capacités, ses buts ainsi que ceux des autres agents. Ce modèle est fondé sur des extensions de la logique, et est basé sur les états mentaux suivants :

- **Les croyances** : les connaissances de l'agent sur l'environnement,
- **les désirs** : les « options » de l'agent, c'est à dire les différents objectifs vers lesquels l'agent peut vouloir transiter,
- **Les intentions** : les états vers lesquels l'agent a choisi de transiter.

L'agent cognitif traite généralement des informations qualitatives tout en utilisant un raisonnement qualitatif ou symbolique.

3.2 Agents réactifs

Selon Brooks, le comportement intelligent devrait émerger de l'interaction entre divers comportements plus simples. [BRO 86 ; 91]

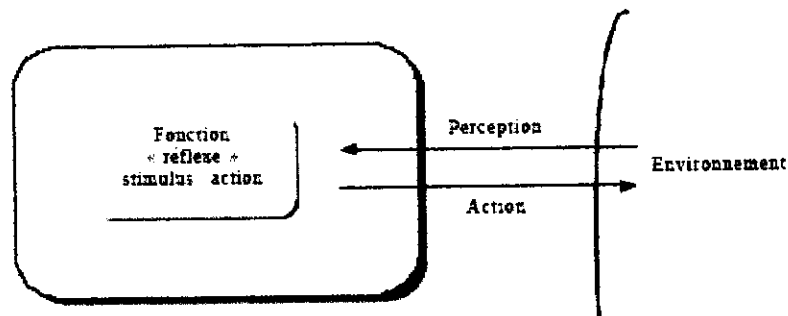


Figure 1.7 : Modèle d'un agent réactif

Dans ce même contexte, nous considérons que les agents réactifs n'ont de connaissance explicite, ni de l'environnement, ni des autres agents, ni de leur passé, pas plus que de leurs buts (pas de planification de leurs actions). Ce sont des agents qui réagissent uniquement à leur perception de l'environnement et qui agissent en fonction de cette perception (voir figure 1.7).

Ce modèle d'agent est une métaphore du modèle « fourmi », il s'appuie sur les sciences de la vie et l'intelligence collective [BON 94 ; 99].

Les agents réactifs, traitent généralement des informations quantitatives tout en utilisant des calculs élémentaires ou d'optimisation.

Les agents réactifs sont les plus simples à implémenter, et leur efficacité a été vérifiée dans plusieurs applications. Ils présentent toutefois des limites dues aux points suivants :

- L'agent n'a pas la représentation mentale de l'environnement, et doit choisir ses bonnes actions à partir des données locales uniquement.
- Le comportement global de l'agent ne peut pas être facilement prévu, par conséquent, il n'est pas toujours possible de concevoir un comportement d'agent en fonction du but spécifié.

Contrairement aux agents réactifs, les agents cognitifs sont beaucoup plus complexes et plus difficiles à mettre en oeuvre. Chaque agent se fonde sur ses propres compétences de façon isolée pour résoudre un problème, ce sont les principales limites de cette architecture.

Pour faire face aux inconvénients de ces deux modèles, les chercheurs ont combiné ces deux facettes, opposées mais complémentaires, de la conception des agents. Cette combinaison fait apparaître les agents hybrides.

3.3 Agents hybrides

Les agents hybrides sont conçus pour combiner des capacités réactives à des capacités cognitives, ce qui leur permet d'adapter leur comportement en temps réel à l'évolution de l'environnement [FER 92 ; FIS 99]. Dans le modèle hybride, un agent est composé de plusieurs couches, rangées selon une hiérarchie. La plupart des architectures considèrent que trois couches suffisent amplement. Ainsi, au plus bas niveau de l'architecture, on retrouve habituellement une couche purement réactive, qui prend ses décisions en se basant sur des données brutes en provenance des senseurs. La couche intermédiaire fait abstraction des données brutes et travaille plutôt avec une vision des connaissances de l'environnement. Finalement, la couche supérieure se charge des aspects sociaux de l'environnement (communication, coopération, négociation), c'est-à-dire du raisonnement tenant compte des autres agents.

Le modèle InteRRap proposé par Müller et Pischel, voir figure 1.8 [MUL 94] figure parmi les exemples du modèle hybride. Dans ce modèle, chaque couche comporte une base de connaissance en fonction de son degré d'abstraction : modèle de l'environnement au niveau de la couche réactive, modèle mental au niveau de la couche de planification locale et modèle social pour la couche de coopération et de planification globale.

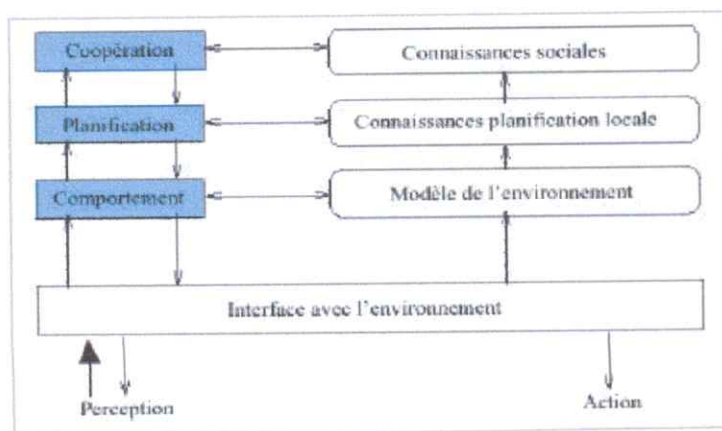


Figure 1.8: Modèle d'agents InteRRap

4. Définition d'un système multi agents (SMA) :

Un système multi agents est une communauté d'agents autonomes en interaction. Nous allons formaliser ceci en présentant des définitions plus connues de la littérature.

Selon [FER 05], un SMA est défini comme:

- Un ensemble B d'entités plongées dans un Environnement E (E est caractérisé par l'ensemble des états de l'environnement S).
- Un ensemble A d'agents avec $A \cap B$.
- Un système d'action (opérations) permettant à des agents d'agir dans E (une opération est une fonction de $S \Rightarrow S$).
- Un système de communication entre Agents (envoi de messages, diffusion de signaux,...).
- Une organisation O structurant l'ensemble des agents et définissant les fonctions remplies par les agents (notion de rôle et éventuellement de groupes).
- Eventuellement : une relation à des utilisateurs U qui agissent dans ce SMA via des agents interfaces $U \cap A$.

Ainsi, cette définition va être résumée dans la figure qui suit (figure 1.9).

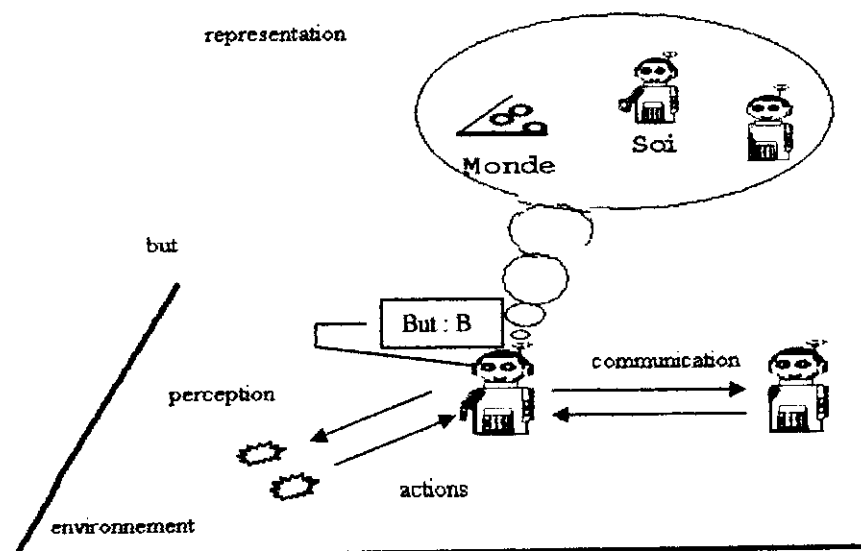


Figure 1.9 : Notion de système multi agents [FER 95]

D'un autre côté [CHA 01] propose une autre définition : « *Un système multi agents (ou SMA) est un système distribué composé d'un ensemble d'agents, interagissant selon des modes de coopération, de concurrence et de coexistence. Un SMA se distingue d'une collection d'agents indépendants par le fait que les agents interagissent en vue de réaliser conjointement une tâche ou d'atteindre conjointement un but particulier.* »

4.1. Organisation :

Le concept d'organisation est vaste mais on peut tout de même le définir comme étant une structure décrivant les interactions et autres relations qui existent (dans le but d'assouvir un objectif commun) entre les membres de la dite organisation. L'organisation peut donc apparaître comme une structure de coordination et de communication [MAL 87]. Jacques Ferber [FER 95] va dans ce sens en affirmant que les organisations constituent à la fois le support et la manière dont se passent les inter relations entre les agents, c'est-à-dire dont sont réparties les tâches, les informations, les ressources et la coordination d'actions. Il précise que ce qui rend l'organisation si difficile à cerner est qu'elle est à la fois le processus d'élaboration d'une structure et le résultat de ce processus. Cette citation met en avant la dualité du terme organisation : en effet, il y a l'aspect statique et l'aspect dynamique. Dans [HUB 02] l'organisation est décrite via une spécification structurelle, une spécification fonctionnelle et une spécification déontique.

L'aspect dynamique, appelé aussi plus simplement organisation, est relatif aux relations entre les éléments du système, qui sont soumis à des changements dynamiques. L'aspect statique, appelé structure organisationnelle provient de la raison d'être des organisations qui est la nécessité pour des individus de se regrouper pour repousser leurs propres limites, en terme de capacités [MAR 58]. Il existe de nombreuses structures organisationnelles [BAE 96]. On en dénombre essentiellement trois grandes familles (les groupes, les hiérarchies et les marchés) basées sur les trois processus de base de coordination [AGI 96] étudiés par Mintzberg [MIN 82] et qui sont :

- l'ajustement mutuel lorsqu'un ou plusieurs agents se mettent d'accord entre eux pour partager une ressource afin de réaliser un but qui leur est commun.

- la supervision directe qui fait intervenir une relation hiérarchique entre les agents : le gérant grâce à son contrôle sur les autres agents peut réguler la consommation faite par les agents qui sont soumis à son autorité.
- La standardisation où l'agent gérant, qui a autorité sur les autres, met en place des procédures à réaliser par les autres agents dans des cas concrets.

4.2. Les types de SMA :

FERBER a cité deux types de SMA [FER 95] :

a) SMA purement communicants :

C'est un système multi agents dont les agents communiquent directement entre eux par le biais de messages. Lorsqu'il n'y a pas d'environnement et que les agents ne font que communiquer, ces systèmes sont très courants en intelligence artificielle distribuée.

Leur domaine de prédiction est la coopération de modules logiciel dont la fonction est de résoudre un problème ou d'élaborer une expertise (interprétation de signaux ou conception de produit par ex) à partir de module spécialisés.

Ces systèmes se caractérisent par le fait que les interactions sont essentiellement des communications intentionnelles et que le mode de travail ressemble à celui d'un organisme social (groupe de travail, entreprise, administration...)

La plupart des SMA réactifs considèrent que la notion d'environnement est fondamentale pour la coordination des actions entre plusieurs agents.

b) SMA situé

Il s'agit d'un SMA dont les agents sont positionnés dans un environnement, et SMA purement situé pour désigner un SMA situé dans lequel les agents ne communiquent pas par envoi de messages, mais seulement par propagation de signaux .

- La dualité agent /environnement est au cœur des SMA située en agissant à partir de ses perceptions de l'espace physique et des communications directe qu'il reçoit . l'agent se définit comme l'image duale de son environnement, c est à dire par ce qui le distingue de ce qui l'entoure.
- Inversement, l'environnement d'un agent est caractérisé par tout ce qui n'est pas lui

4.3. Formes d'allocation de tâches dans un SMA:

La gestion de la répartition des tâches peut s'effectuer soit en centralisant le processus d'allocation, soit en le distribuant à l'ensemble des agents concernés.

Dans un mode d'allocation centralisé, deux cas se présentent [FER 95] :

1. si la structure de subordination est hiérarchique, c'est le supérieur qui de mande précisément à un subordonné d'accomplir sa tâche « allocation définie ».
2. au contraire si la structure est égalitaire, la répartition passe alors par la définition d'agents spéciaux « les médiateurs ».

Dans un mode d'allocation distribué, deux cas se présentent :

1. le mode d'allocation par réseau d'acointances suppose que les clients possèdent une représentation des autres agents et des capacités dont ils disposent.
2. le mode d'allocation par appel d'offre est plus connu en IAD sous le nom de réseau contractuel, il présente l'avantage d'une très grande dynamique et s'avère particulièrement facile à mettre en œuvre.

4.4. Interaction dans un SMA

Un système multi agents constitué d'agents cognitifs se base essentiellement sur l'interaction entre agents. L'interaction a comme objectif de mettre en relation dynamique deux ou plusieurs agents par le biais d'une ou plusieurs actions réciproques. Elle peut se faire selon plusieurs manières suivant la situation à travers laquelle interagissent nos agents.

On distingue différents types d'interaction que les agents peuvent adopter comme: la communication, la coopération, la négociation et la coordination. L'interaction se base sur des protocoles qui permettront aux agents d'échanger des messages structurés.

Elle a comme conséquence de mettre en question les intentions d'un agent. En effet, l'agent doit s'adapter aux changements subis par son environnement et par les autres agents. Ces changements vont avoir des répercussions sur son comportement.

Le fait d'interagir va permettre à l'agent de partager de l'information, d'être au courant de ce qui se passe autour de lui, d'atteindre ses buts et d'éviter autant que possible les conflits.

Et comme l'affirme [FER 95] : « *Un agent sans interaction avec d'autres agents n'est plus qu'un corps isolé, qu'un système de traitement d'information, dépourvu de caractéristiques adaptatives* ».

Dans ce qui suit, nous détaillons les différents modes d'interaction qui sont évoqués dans cette partie et les protocoles sur lesquels elle se base.

1. La communication :

La communication est un des éléments importants du système multi agents. Elle permet l'échange des informations, la coopération et la coordination. Dans la communication agent, l'intention de communiquer est de produire un effet sur les destinataires : ils exécuteront probablement une action demandée par l'émetteur ou répondront à une question [HUG 05].

La communication entre agents peut être une communication directe ou indirecte.

La communication indirecte est adoptée par les agents réactifs. Elle se fait à travers l'environnement et donc elle est non intentionnelle, ou bien par le biais d'un tableau noir est, dans ce cas, intentionnelle. Dans une communication par environnement, les agents laissent des traces ou des signaux qui seront perçus par les autres agents. Dans ce genre de communication il n'y a pas de destinataire bien défini.

La communication directe, quand à elle, est spécifique aux agents cognitifs. Elle est intentionnelle et se fait par envoi de messages à un ou plusieurs destinataires. Dans ce qui suit nous allons détailler les différents aspects de la communication.

1.1. La communication indirecte

La communication indirecte peut se faire soit par :

- *Environnement* : les agents dans ce genre de communication, laissent des traces ou des signaux qui seront perçus par les autres agents.
- *Partage d'informations* : C'est la notion de tableau noir (Blackboard). En fait, les agents déposent et lisent les informations dans une zone de données communes appelée aussi mémoire partagée. Chaque agent filtre les informations quand il accède au tableau pour trouver ce qu'il lui faut. Pour se faire, un agent doit impérativement s'inscrire à un site central pour avoir l'autorisation d'écrire sur le tableau noir.

1.2 La communication Directe

Elle se fait par envoi de messages : un agent peut envoyer des messages à un ou plusieurs agents. On appelle communication point à point quand elle se fait entre deux parties. Quand la communication se fait d'un agent vers plusieurs agents ou de plusieurs à plusieurs, elle est qualifiée de communication par multi parties (Multi-party communication) [HUG 05].

La communication directe se base sur trois éléments essentiels :

a) Le langage de communication :

Afin de pouvoir communiquer et se comprendre, les agents ont besoin d'un langage commun. Jusqu'à maintenant, les protocoles de communication les plus utilisés et les plus connus sont KQML (Knowledge Query and Manipulation Language) et FIPA-ACL (Foundation for Intelligent Physical Agent-Agent Communication Language). Ils sont basés sur la théorie des actes de langages [AUS 62] et [SEA 69] : les messages sont des actions ou des actes communicatifs.

KQML est le premier standard de communication entre agents qui a été développé en 1993 par Finin et al pour la DARPA (Defense Advanced Research Project Agency). C'est un protocole qui est destiné aux agents intentionnels et qui se base sur le principe que tout ce qui est nécessaire à la compréhension du message est inclus dans le message lui-même. C'est un langage de communication de haut niveau orienté message et un protocole pour l'échange des informations indépendamment de la syntaxe du contenu ou de l'ontologie de l'application.

KQML sépare la sémantique du protocole de la sémantique du message. Il est composé de 35 performatifs (17 performatifs de discours, 11 performatifs d'interconnexion et 7 performatifs d'exception) [HUG 05]. Un message KQML se compose alors d'un performatif (action à faire) qui est l'en-tête du message et d'un certain nombre de couples attributs / valeurs.

Il a donc la forme suivante :

```
(KQML-performative
: sender <word> //l'émetteur du message
: receiver <word> //le destinataire du message
: language <word> //le langage d'expression du message
: ontology <word> //le vocabulaire du domaine
: content <expression>... ) //le contenu du message
)
```

Les autres arguments du langage sont :in-reply-to, :in-reply-with, :from, :to.

L'avantage de KQML est qu'il est supporté par beaucoup d'applications, et qu'il est extensible (nouveaux performatifs, nouveaux paramètres, nouvelles ontologies). Ceci dit KQML tend à être remplacé par FIPA-ACL (Agent Communication Language) pour la clarté de sa sémantique, son support de communication, les protocoles, etc.

- ***Les protocoles de FIPA***

FIPA spécifie un environnement d'existence et de fonctionnement des agents ainsi qu'une infrastructure physique de déploiement des agents [FIPA 97].

FIPA définit une plate-forme des agents qui représente l'infrastructure dans laquelle les agents peuvent être déployés, un système de gestion des agents décrit tel les agents peuvent gérer la création, la suppression, la suspension, la reprise, l'authentification et la migration des autres agents tout en assurant un service des pages blanches ainsi qu'un service des pages jaunes [M.FA 02].

Nous allons présenter ci-dessous quelques protocoles d'interaction de FIPA :

- ✚ Le protocole FIPA-request : il permet à un agent d'inviter d'autres agents à exécuter une certaine action. L'agent récepteur peut refuser ou accepter la tâche. S'il accepte, il doit exécuter l'action et informer l'agent qui l'a demandée.
- ✚ Le protocole FIPA-query : l'agent cherche à connaître un objet correspondant à une description ou interroger un autre agent sur la proposition ou l'état sur l'exécution d'une autre action.

- ✚ Le protocole FIPA-contract-net : l'agent prend ici le rôle du manager, le manager souhaite faire accomplir une certaine tâche par un ou plusieurs autres agents, et optimiser une fonction qui caractérise la tâche.
- ✚ La protocole FIPA –Iterated-contract-net : c'est une extension du protocole précédent, il diffère de la version de base en permettant plusieurs itérations. Une fois que l'agent manager reçoit les propositions des contractants, il peut accepter une ou plusieurs offres, rejeter d'autres ou peut retirer le processus en émettant une nouvelle offre révisée.

En 1999, FIPA a développé un nouveau ACL qui ressemble beaucoup à KQML. FIPA-ACL est basé sur les actes de communication qui sont décrits sous une forme narrative et une sémantique formelle.

Le performatif *inform* est le plus important dans les messages du langage FIPA-ACL. Il permet à un agent émetteur d'envoyer une information qu'il croit vraie.

La sémantique de ACL-FIPA veut alors que l'agent receveur la met dans sa base de connaissance c'est-à-dire que lui aussi la croit vraie.

La syntaxe FIPA-ACL est la même que KQML, donc pour cette raison qu'on ne va pas présenter un exemple vu que c'est la même structure que celui présenté avec KQML.

b) L'Ontologie :

Dans les dictionnaires du Net, définit une ontologie comme étant « une organisation hiérarchique de la connaissance sur un ensemble d'objets par leur regroupement en sous catégories suivant leurs caractéristiques essentielles » [TOU 06].

Cette définition générale nous amène à mieux comprendre la définition proposée par Gomez-Perez, Corcho et Fernandez-Lopez en 2003 : « une ontologie définit les termes de base et les relations comprenant aussi bien le vocabulaire du domaine concernée que les règles pour combiner les termes et les relations afin de définir des extensions au vocabulaire ».

Ainsi, l'ontologie servira à fournir un vocabulaire et une terminologie compréhensible par tous les agents. Cette sémantique sera régie par des règles et des contraintes qui permettront de définir un consensus sur le sens des termes. En effet, l'ontologie concernera les symboles non logiques du contenu du message.

c) Support de communication :

La communication nécessite un support qui est un mécanisme permettant de stocker, rechercher et adresser les messages aux agents. Ces mécanismes sont présents dans les plateformes multi agents comme Jade ou MadKit.

1.2.1 Structure de la communication directe

Dans les systèmes multi-agents il existe trois approches pour structurer la communication directe entre agents cognitifs [TOU 06] : les protocoles, les jeux de dialogue et les systèmes d'argumentation.

a). *Les protocoles :*

Parmi les protocoles les plus utilisés et les plus connus dans le système multi agents, on trouve le réseau contractuel "Contract Net" qui décrit un protocole dont les agents coordonnent leurs activités en établissant des contrats afin d'atteindre leurs buts.

Nous nous intéressons à ce protocole vu que c'est un processus de contrôle simple à comprendre et à utiliser. Le réseau contractuel est un mécanisme d'allocation de tâches fondé sur la notion d'appel d'offre.

L'appel d'offre s'effectue en 4 étapes [FER 95] :

1. l'administrateur envoie une description de la tâche qu'il voudrait voir effectuée à tous ceux qu'il estime pouvoir répondre ou à tous les agents du système.
2. les offrants élaborent une proposition qu'ils envoient à l'administrateur.
3. l'administrateur reçoit et évalue les propositions, attribue le marché au meilleur offrant au cours de la troisième étape.
4. l'offrant qui a reçu le marché devient un contractant.

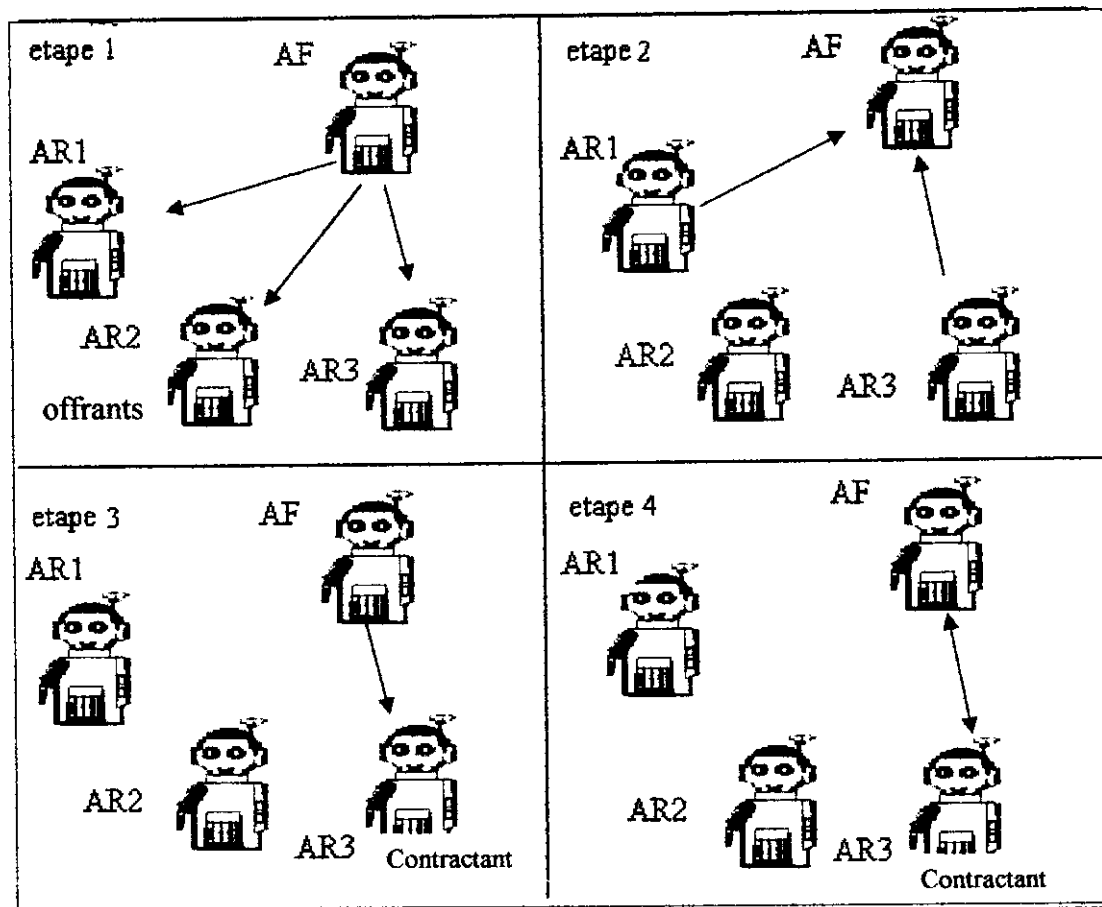


Figure 14 : les quatre étapes du réseau contractuel

b) *Les jeux de dialogue :*

Cette théorie considère les agents comme des joueurs.

L'interaction se fait entre un ou plusieurs joueurs selon des règles de jeu prédéfinies auparavant. Dans les jeux de dialogue, un seul type de communication est défini : pour chaque requête une réponse est requise. Dans la littérature, il existe différents types de dialogues :

1. Recherche d'information : c'est le cas où un agent a besoin de demander des informations à un autre agent en sachant que cet agent pourra lui répondre.
2. Enquête : les agents collaborent afin de répondre à une ou plusieurs questions inconnues par tous les participants.
3. Persuasion : c'est le cas d'un participant qui veut persuader un autre participant à accepter sa proposition qu'il n'a pas encore approuvé.

4. Négociation : c'est une négociation autour d'une ressource acceptée par tous les participants mais qui est en conflit avec l'objectif individuel de chacun.
5. Délibération : c'est lorsque les agents collaborent pour décider de l'action à entreprendre pour une situation donnée.

c) *Systeme d'argumentation* :

Les protocoles et les jeux de dialogue considèrent que les informations provenant des agents sont consistantes et toujours vraies. Le système d'argumentation quant à lui, met en doute les croyances de l'agent et ce dernier doit appuyer ses croyances par des informations supplémentaires.

2. La coopération :

La coopération se traduit par le fait qu'un ensemble d'agents travaillent ensemble pour satisfaire un but commun ou individuel. L'ajout ou la suppression d'un agent influe considérablement sur la performance du groupe.

Le besoin de faire coopérer des agents, vient essentiellement du fait qu'un agent ne peut atteindre son objectif individuellement et , par conséquence besoin de l'aide des autres agents du système. Quelle que soit l'organisation d'une société d'agents, un agent peut cooperer suivant les modes suivant les modes suivants :

coopération par partage de taches et de résultats :

- *Par partage de taches* : Le problème est distribué entre les différents agents travaillant indépendamment les uns des autres ; chaque agent dispose des ressources et des compétences nécessaires pour accomplir la tache qui lui a été assignée.
- *par partage de résultats* : les agents ne peuvent accomplir leurs taches de manière indépendante. Ils sont appelés à se transmettre mutuellement des résultats partiels.

coopération par commande :

Dans ce mode, un agent superviseur décompose la tache à effectuer en sous taches qu'il répartit entre les autres agents qui lui paraissent les plus appropriés pour la résolution.

coopération par compétition :

Un agent A décompose et diffuse la liste de sous problèmes comme le mode d'appel d'offre, chaque agent résout un ou plusieurs sous problèmes et envoie les résultats correspondants à A qui à son tour fait le tri.

coopération par appel d'offre :

Nous avons vu ce mode dans la structure de communication directe.



3. La coordination :

La coordination est une propriété importante dans les systèmes multi agents. Elle est présente lorsqu'il existe une interdépendance dans les actions des agents, leurs buts ou même les ressources qu'ils utilisent. A ce stade, coordonner les activités des agents devient un aspect essentiel afin d'éviter des problèmes de collision ou de retard qui auront des répercussions considérables sur l'objectif global du système en question [TOU 06].

En effet, la coordination met de l'ordre dans le processus de travail des agents en répartissant les tâches. On évite ainsi de faire installer le chaos vu que chaque agent a une vue locale de son environnement et ne peut pas connaître tout le système. Donc la coordination consistera à aider les agents à atteindre le but global du système en synchronisant leurs activités ou en réglant les conflits qui existent entre eux. A partir de ceci, nous remarquons que la coordination est une propriété indispensable dans un système multi agents et qu'elle est utilisée comme une manière d'interagir qui consolide la coopération entre les agents ou aussi la négociation.

4. La négociation

Un système multi agents est composé de plusieurs agents. Ces derniers peuvent entrer en conflits pour plusieurs raisons : conflits d'intérêts ou de buts, accès à des ressources ou proposition de plusieurs solutions différentes à un seul problème.

Afin de résoudre ces conflits et de trouver une situation qui satisfasse tout le monde, les agents négocient entre eux en faisant des concessions ou en cherchant des alternatives [TOU 06].

La négociation peut être ainsi résumée en un processus d'échange entre agents qui résulte de la planification et du raisonnement. Chaque agent a une meilleure connaissance sur ce que veulent faire les autres agents et à partir de ceci réajuster ses propres plans ou voir même influencer le comportement des autres agents.

5. Agent Oriented Programming (AOP) :

Le paradigme AOP [SHO 93] propose un système constitué d'un langage formel pour décrire les états mentaux d'un agent d'un langage de programmation interprété dont la sémantique correspond à celles des états mentaux d'un agentifieur traduisant un logiciel en un agent fonctionnant avec un ensemble de capteurs et d'effecteurs réels, du même ordre que ce qui a été proposé pour les automates situés [ROS 86].

L'architecture proposée par Briot et Demazeau [BRI 01] est décrite par les primitives présentées au sein de l'état mental. Ce sont des croyances, les compétences, les choix et les engagements.

La première réalisation du langage correspondant (Agent Oriented), permet de représenter des faits, des actions conditionnelles ou inconditionnelles (Do, Refrain) internes ou de communication, et des règles d'engagement qui décrivent la manière dont l'agent s'engagera en fonction de son état mental et des messages sont seulement été définis pour mettre à jour les croyances de l'agent (Inform), demander (Request) ou annuler une demande (Unrequest) en mettant à jour les engagements.

Le langage de programmation est constitué d'un interpréteur qui met en œuvre un cycle de contrôle constitué de deux étapes :

Mise à jour des états mentaux à partir des messages reçus, génération et exécution des engagements courants définis à partir des règles d'engagements. Cette première version très simple et rudimentaire du langage a été étendue dans PLACA (Planning Communicating Agents) [THO 95].

Les ajouts ont concerné les communications et le raisonnement sur les buts, la planification. La modalité d'Intention a été introduite et le cycle de l'interpréteur a été modifier en y introduisant des étapes, dépendantes du temps, de construction et de raffinement de plan, une architecture similaire à AOP sert d'élément de base dans la plate-forme KAOS [BRA 96] avec l'adjonction d'un module explicite de gestion des conversations.

6. Les SMA et la coopération des SI

Les systèmes d'information coopératifs centrés sur la notion d'agent, plus précisément d'agent informationnel (AI) conduisant au concept de SI orientés agents (AOIS); actuellement en émergence dans la communauté scientifique.

Nous appréhenderons ces SIC centrés agents ou AOIS selon deux voies différentes associées à la stratégie d'introduction de cette notion d'agent dans les SI.

- La première consiste à « agentifier » le SI. Elle découle d'une évolution de l'ingénierie classique des SI et plus particulièrement de l'ingénierie de BD distribuées et fédérées. Ainsi que de l'usage récent de la technologie agent dans les BD actives et dans la coopération entre plusieurs sources d'information hétérogènes.
- La seconde voie nous vient de l'IAD. Plus particulièrement des SMA, elle appréhende le SI comme un SMA composés d'agents informationnels qui ont à coopérer afin d'exploiter ou de coordonner différentes sources d'informations hétérogènes. C'est cette deuxième voie qui nous intéresse dans le cadre de notre travail.

6.1. Agent informationnel

Un agent informationnel peut être appréhendé comme un ensemble de modules logiciels disposant de compétences particulières et transformant les SI composants, par nature passifs, en composants actifs capables de raisonner sur leurs fonctionnalités et leurs compétences et d'interagir pour résoudre un problème donné. Selon [PAP 92], pour assurer une recherche et une intégration intelligente d'information. Les AI doivent posséder les fonctionnalités suivantes :

- 🔗 Traduction d'information : cette fonctionnalité permet à l'AI homogénéiser la représentation des données, les formats et les langages de différents SI hétérogènes. Conformément aux conventions d'une organisation
- 🔗 Intégration partielle des informations : l'AI doit pouvoir intégrer partiellement des informations issues des différents sources et sémantiquement reliées.

- ✚ Autoreprésentation : les AI doivent pouvoir s'autodécrire évoluer dynamiquement et être reconfiguration afin de s'adapter à l'évolution des sources d'information.
- ✚ Communication expressive avec les autres agents : des protocoles et des langages de communication sont indispensables pour traiter les requêtes et pour propager leurs information.
- ✚ « courtage » de l'information : cette fonctionnalité permet a l'AI de rechercher et de localiser des sources d'informations, des services, et d'autres AI susceptibles de participer à la résolution d'une requête posée par l'utilisateur.

6.2. Agent Informationnels pour la recherche et l'intégration de l'information :

Pour faire face à l'explosion de l'information disponible sur les réseaux, due en particulier au développement de l'Internet et du web, des moteurs et des systèmes sont actuellement proposés. Ces moteurs comme par exemple AltaVista sont basés sur la fréquence de mots-clés dans les documents et fournissent, en réponse à une requête posée par l'utilisateur une liste d'URL (Universal Ressource Location) souvent conséquente et pas toujours pertinente.

Des moteurs de recherche de plus haut niveau, appelés moteurs de metarecherche. Ces derniers construisent à partir des documents trouvés par le moteur de recherche, des groupes de documents en fonction de critères tels que la similarité des documents ou la redondance des URL.

Des lors ces moteurs fournissent également trop de données à l'utilisateur. Dans ce contexte, la technologie d'agent à base d'agents informationnels (AI). Commence à être utilisée pour supporter une recherche et une intégration intelligente d'information, nécessaires notamment dans le domaine du commerce électronique.

6.3. Fonctionnalités des AI pour la recherche et l'intégration d'information :

Les systèmes de recherche et l'intégration d'information basés sur des AI, ont pour rôle d'extraire l'information de documents qu'ils soient structurés ou non, de planifier la recherche de l'information, d'intégrer l'information pour supporter un processus de décision, de raisonner sur les échanges entre les ressources informationnelles, et d'utiliser l'information extraite pour raffiner la recherche et traiter l'information.

Ainsi pour pallier les faiblesses des moteurs de recherche [AGE 2], plutôt que de consulter un grand nombre de sites, parcourir les liens pour trouver des informations relèvent du domaine. Ils utilisent des règles, des questionnaires ou des techniques d'apprentissage. Ils ne sont pas aussi rapides que les moteurs de métarecherche mais ils privilégient la qualité des documents et permettent d'obtenir des documents qui relèvent mieux du domaine concerné.

6.4. Principaux concepts, techniques et langage utilisés :

Afin d'assurer les fonctionnalités décrites précédemment, des concepts, des techniques et des langages sont souvent utilisés par les systèmes de recherche et d'intégration d'information

Une ontologie peut être définie comme un vocabulaire spécifique et des relations utilisés pour décrire certains aspects de la réalité, et un ensemble de postulats liés au sens du vocabulaire des mots [GUA 98].

Les ontologies sont souvent utilisées dans les AI, d'une part dans la fonction de traduction pour fournir les bases d'une terminologie commune et faire correspondre les différentes terminologies utilisées dans les ressources, d'autre part, dans les fonctions d'intégration, de courtage pour modéliser le modèle d'expertise et de communication entre les agents.

- Le contexte d'une requête d'information peut par exemple aider le système à distinguer si le terme main levée relève de l'action de lever la main de l'acte juridique, le contexte prend en compte les métadonnées, les profils d'utilisateurs et les ontologies.
- Les protocoles de communications entre AI proposent des primitives de communication. Ils sont souvent basés sur la négociation est dotés de capacités argumentatives et explicatives. La négociation est nécessaire lorsque des problèmes de résolution de conflits se posent du fait de l'hétérogénéité des SI composants.

- En effet les agents peuvent être en concurrence pour obtenir des ressources, des conflits peuvent apparaître dans la phase de l'intégration de l'information issue de différentes sources d'information.
- Pour réaliser une communication expressive et la négociation entre les agents, des recherches ont concentré leurs efforts sur des techniques de partage de la connaissance et on donné lieu à des langages comme KQML. »

Conclusion :

Les SMA s'avèrent d'être des outils très intéressants à plusieurs titres [ESP 98] :

- Par leur puissance en tant qu'outil de modélisation de connaissances plus précisément leur capacité à décrire des domaines complexes et leur souplesse permettant la mise à jour des modèles.
- par leur capacité d'exécution et /ou de simulation permettant la validation de modèle.
- par leur capacité à appréhender des notions telles que la distribution, la coopération, la coordination et la négociation.

En conséquence, les SMA ont été largement utilisés en tant qu'outil de modélisation pour décrire des systèmes industriels complexes [CAM 90, JEN 98].

Ils ont aussi donné lieu à un nombre croissant d'applications industrielles opérationnelles apportant des solutions originales et efficaces notamment dans les domaines de la gestion de production, de l'ingénierie simultanée du pilotage ou encore de la gestion de SI [CHA 94].

Dans ce chapitre, nous avons fait le tour de la technologie agent et des systèmes multi agents. En effet, nous avons essayé de présenter les points les plus importants des agents pour pouvoir les utiliser par la suite dans notre système.

Nous avons abordé dans la seconde partie l'utilisation des systèmes multi agent dans la coopération des SI, en effet, grâce aux caractéristiques propres des agents au niveau de la modularité, la décentralisation, l'adaptabilité, nous sommes en mesure de justifier l'utilisation des agents.

Vis-à-vis de nos préoccupations en matière de coopération de SI à base d'agent les problématiques que nous retenons sont les suivantes :

- Comment faire en sorte que l'organisation de SMA soit la projection naturelle de l'organisation de l'entreprise orientée d'avantage vers le client et sujette à une perpétuelle évolution ?
- Quelles sont la forme et la fonction concernant un agent sachant que le comportement doit dépendre de cette forme et de cette fonction ?
- Comment rechercher de façon transparente la ressource adéquate dans un environnement ouvert et dynamique ?
- Comment négocier avec le serveur pour obtenir le service le plus adapter ?
- Quelles sont les formes d'interactions permettant aux agents de trouver conjointement une réponse à une requête d'un utilisateur ?

Pour modéliser le SI coopératif à base d'agents, nous avons opté pour AUMML comme langage de modélisation ; une brève description de ce langage est donnée au chapitre suivant.

Chapitre III :

Langage de modélisation AUML

Introduction :

La modélisation est une des tâches les plus importantes dans le processus de développement d'un système. La phase consacrée à l'analyse peut être considérée comme plus stratégique que celles dévolues à la conception et l'implémentation proprement dites. Il faut en effet fondamentalement représenter, comprendre et identifier les exigences du système afin de concevoir puis d'implémenter une application stable et performante.

Pour la modélisation de notre système nous avons choisi le langage Agent UML c'est une extension de la notation UML.

Agent-UML est un langage de modélisation oriente agents. Il a pour but d'étendre UML avec des moyens adaptés aux Systèmes multi agent [ODE 00]. Ces derniers ont été souvent caractérisés comme une extension des systèmes orientés objets.

Agent-UML a comme but de recommander une technologie pour l'adoption d'une sémantique, d'un méta modèle et d'une syntaxe abstraite standard pour l'analyse et la conception des méthodologies basées sur les agents. Cette technologie devra permettre de couvrir le cycle de vie des produits et des outils de travail AUML et être en accord avec les spécifications existantes de FIPA (Foundation for Intelligent Physical Agents) et OMG (Object Management Group).

Les notations proposées sont :

- Une représentation des processus simultanés d'interaction permettant ainsi à UML de modéliser les protocoles d'interactions entre agents. (par exemple le cas de transmission de messages à plusieurs destinataires/agents)

- La notion de rôle qui permet de modéliser les rôles joués par les agents.

AUML a introduit le diagramme d'interaction qui est une extension de diagramme de séquence d'UML mais avec des opérateurs logiques, de causalité, de synchronisation et de diffusion [ODE. 00]. C'est un diagramme adapté aux protocoles de communication entre les agents

1. Diagrammes de séquence AUML

Le diagramme de séquence AUML est donc très semblable au diagramme de séquence UML. Des opérateurs spécifiques ont été ajoutés notamment pour exprimer le choix etc. Il permet de décrire un protocole d'interaction mettant en jeu divers agents de différents rôles.

Odell *et al.* [ODE 00,01] ont ajouté des types de branchements différents dans les diagrammes de séquence afin de prendre en compte l'indéterminisme du comportement d'un agent – la figure 1.10 montre plusieurs branchements XOR (OU exclusif).

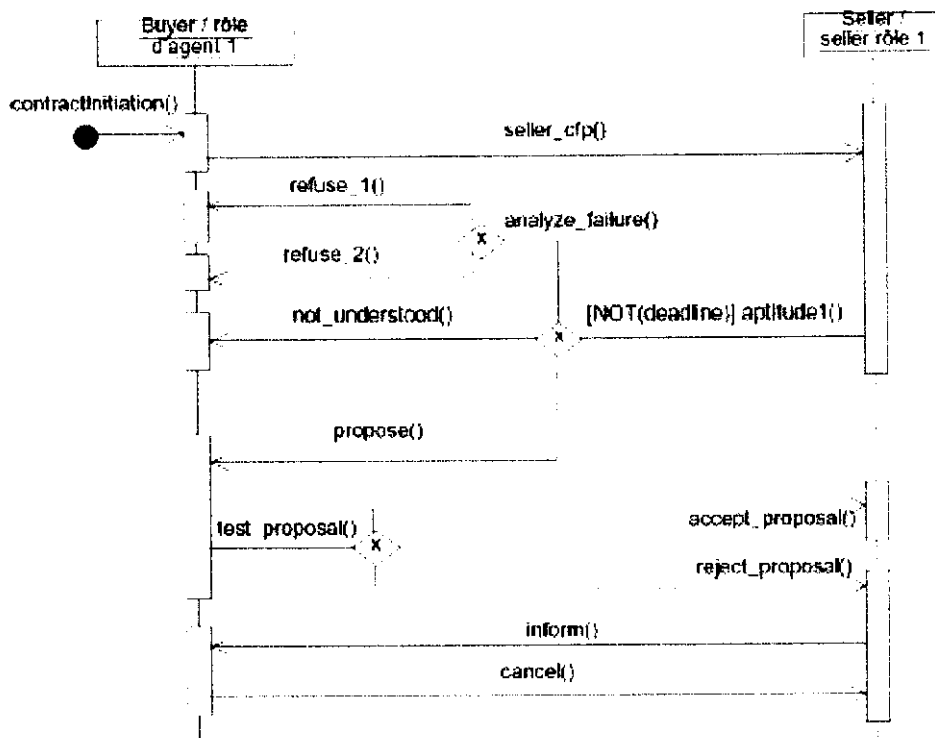


Figure 1.10. – Diagramme de séquence AUML

2. Diagramme de classe

Huget [HUG 02] a proposé une extension des diagrammes de classes UML a fin de prendre en compte les caractéristiques des agents du système, leurs relations (interactions) ainsi que les messages entrants et les messages sortants de la classe, et l'introduction de la notion du rôle d'un agent.

La figure 1.11 illustre une description d'une classe d'agent

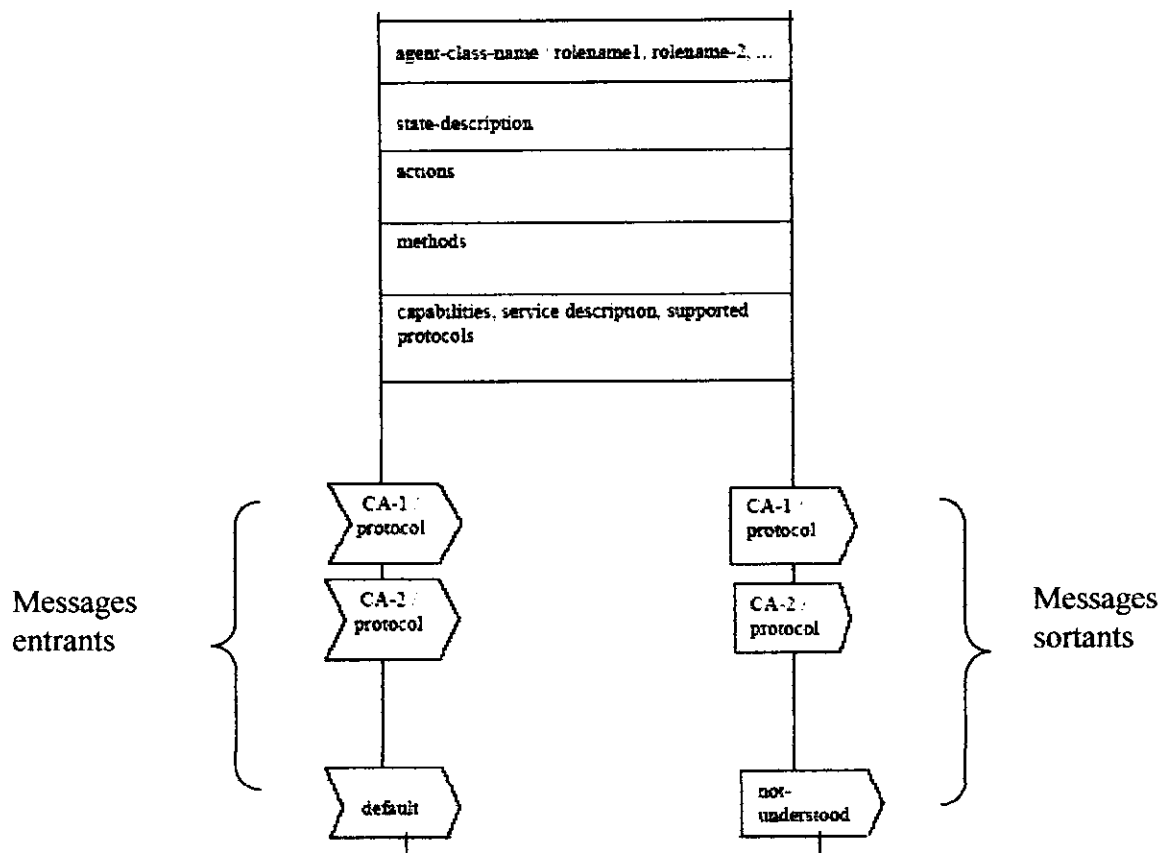


Figure 1.11 : description d'une classe d'agent

Conclusion :

La recherche d'information dans un environnement ouvert et dynamique devient une tâche difficile et les méthodes traditionnelles de recherche sur Internet ou sur des systèmes distribués s'avèrent de plus en plus limitées. Les systèmes d'informations coopératifs basés sur les agents logiciels apportent de solutions prometteuses à cet épineux problème.

Dans la partie « état de l'art » du présent document, nous avons abordé les thématiques fondamentales de cette thèse : les systèmes d'information coopératifs et les systèmes multi agents. Nous avons présenté, dans les chapitres 1 et 2, une vue globale de ces thématiques et de leurs problèmes. Et nous avons réservé chapitre 3 au langage de modélisation AUML.

Nous modélisons le système avec AUML dans la partie suivante.

Deuxième partie

Conception et réalisation du système coopératif

Chapitre I : Conception du système coopératif

Nous considérons dans toute la suite un système d'information distribué coopératif comme le support de toute application permettant, à différentes ressources réparties, de coopérer par le biais d'un système informatique dans le but de réaliser une tâche donnée. Les ressources peuvent être des :

- Programme d'application incluant des données de différents types (bases de données, fichiers, description XML, pages web....).
- Entités applicatives : documents électroniques en circulation, périphériques pilotés par des couches logicielles (télécopieurs, imprimantes....).
- Utilisateurs connectés via un programme d'interface,.....

Ces ressources sont susceptibles d'apparaître, de disparaître ou d'évoluer. Chacune d'elles se présente comme serveur offrant des services et clients cherchant certains services. La notion de coopération correspond à cette interopérabilité intelligente, qui consiste en une démarche active des serveurs et des clients visant une collaboration optimale.

L'objectif de ce travail est de proposer une solution multi agents à la coopération des ressources dans un système d'information distribuée coopératif afin d'assurer un accès transparent aux informations et aux services de cette coopération.

Dans la perspective de développer un SMA au service de la coopération, nous basons notre travail sur le concept de système d'information coopératif proposé dans [OUA 04].

I. Une approche unifiée du SI

L'interopérabilité requise est obtenue en intégrant les ressources du SI coopératif dans une représentation uniforme d'agents autonomes, chargés d'exploiter et de coordonner différentes sources d'information hétérogènes, ces agents ne sont ni purement réactifs, ni purement cognitifs, ceci conduit naturellement à la notion d'agent hybride, et ça nous permet de créer une couche uniforme d'agents résultant de la projection naturelle des entités du monde réel du SI.

Pour cela, nous associons à chaque ressource du système un agent *Ressource* chargé de la représenter vis à vis des autres ressources. L'agent Ressource est lié à la présence de la ressource.

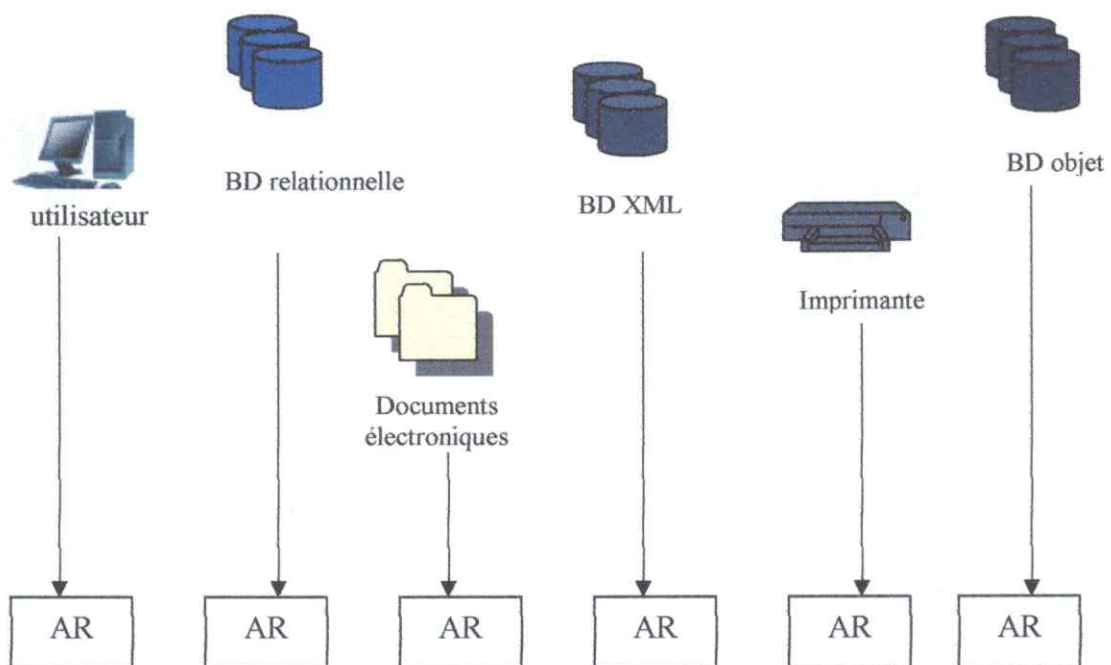


Figure 2.1 : représentation uniforme du système

Au-dessus de ces agents, des agents *fédérateurs* sont définis donnant l'illusion à un utilisateur du système qu'il interroge un système centralisé. L'agent *Fédérateur* regroupe dans un cadre unificateur la description d'un domaine d'intérêt ainsi qu'un ensemble de ressources disponibles relativement à ce domaine. C'est la tâche du *fédérateur* que de prendre en entrée la requête de l'utilisateur et de la transformer en requêtes spécialisées en fonction des informations qu'il possède de ses agents *Ressource*.

La particularité de cette approche est d'offrir une couche homogène d'agents représentant indifféremment les machines, les applications, les sources d'information et les utilisateurs. Les agents *Ressource* représentant les utilisateurs représentent particulièrement la partie intelligente de l'interface homme/Machine, points d'accès à tous les services du système. Chaque agent possède un rôle de négociateur soit en tant que client soit en tant que serveur. Les agents, interlocuteurs exclusifs pour l'accès aux ressources, constituent des sources d'information cohérentes réparties dans l'ensemble du système.

I.1 Organisation du Système Multi Agents

Le SMA est composé d'agents Ressource (AR) et d'agents Fédérateur (AF). Les AF se repèrent par rapport à un réseau d'accointances au sein duquel évolue un ensemble d'agents AR.

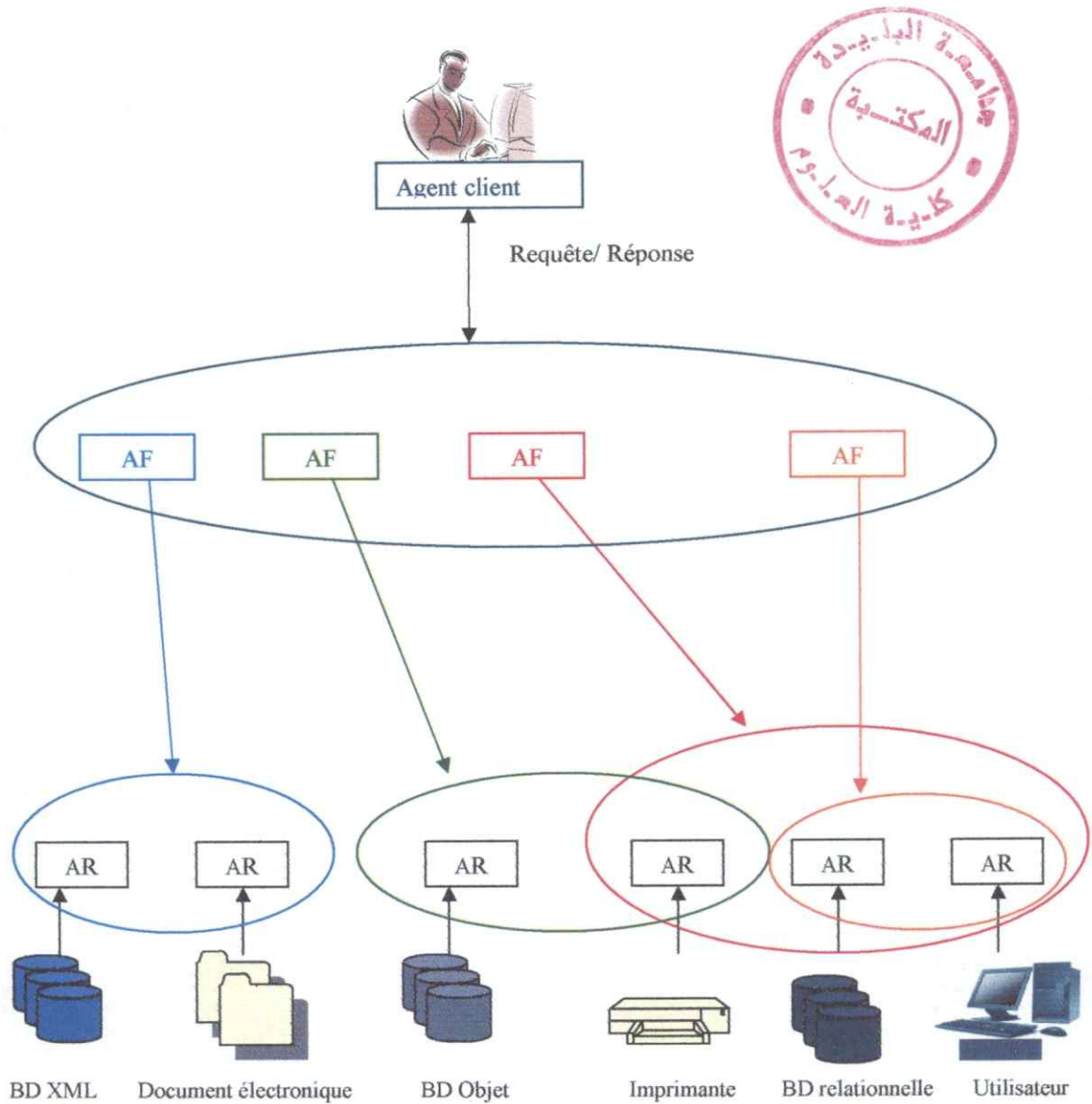


Figure 2.2 : Vue générale du SMA

Les agents AR sont regroupés logiquement en **groupes** pas nécessairement disjoints. Chaque groupe est enregistré sous un nom symbolique attaché à une sémantique partagée par tous les agents susceptibles d'être clients. Un groupe n'est pas connu d'avance par un agent client. La structure mise en place consiste en ce que les agents *ressource* maintiennent une sorte d'association d'un même domaine d'intérêt.

L'association de groupe est maintenue par l'intermédiaire d'un agent *fédérateur* qui contrôle un ensemble de contraintes que doit vérifier l'agent pour qu'il soit répertorié dans le groupe. L'agent fédérateur introduit un niveau d'indirection qui permet aux clients de s'affranchir du nom d'un serveur particulier et leur offre l'opportunité de contacter tous les serveurs du groupe susceptibles de satisfaire la requête.

I.2 Typologie des agents du SMA

Les agents AR disposent de la connaissance sur les ressources du système et les agents AF coordonnent leurs actions en assurant la coopération durant le processus de traitement d'une requête.

➤ Agent Ressource (AR)

Il dispose de compétences particulières transformant la ressource composante, par nature passive, en composant actif capable de raisonner sur ses fonctionnalités et ses compétences et d'interagir pour résoudre une requête donnée. Il se charge :

- **D'accéder** à la ressource pour des accès locaux (requête locale) ou globaux (requête globale),
- **De diffuser** la connaissance présente dans la ressource sous forme d'informations et de services,
- **De faire participer** la ressource au processus coopératif de résolution d'une requête de service global.

Il possède un ensemble de compétences, de connaissances et d'intentions :

- **Compétences** : correspondent à l'ensemble des services offerts par la ressource à la coopération pouvant être invoqués par des agents clients et un ensemble de

services que l'agent sait effectuer de par lui-même sans faire appel au SMA ; il s'agit de services locaux.

- **Connaissances** : concernent les données manipulées par la ressource, l'ensemble des caractéristiques statiques et dynamiques des services offerts, l'ensemble des contraintes associées (règles de gestion, règles d'organisation par exemple) ainsi que la localisation des agents AF auquel il est logiquement rattaché par l'association de groupe.
- **Intentions** : correspondent aux buts à atteindre pour satisfaire une requête formulée par un utilisateur global du système. Ils correspondent aux besoins fonctionnels et non fonctionnels destinés à la description de l'objectif du SI. Si un agent reçoit un message dont le contenu ne correspond pas à ses intentions il le remet sans donner suite.

➤ **Agent Fédérateur (AF)**

Un agent AF n'intervient que pour la réalisation de buts collaborateurs associant un ensemble d'agents. Il possède un ensemble de compétences et de connaissances :

- **Compétences** : nécessaires à la gestion de la coopération :
 - enregistrer les agents AR dans le groupe.
 - diffuser la recherche d'informations et de services auprès des AR.
 - récupérer les résultats issus des AR en fonction des éléments de la requête.
 - décomposer une requête globale en sous-requêtes à affecter à un ensemble d'agents AR.
 - construire une réponse à la requête.
 - négocier, avec l'agent client la requête de service dans le cas où elle ne serait pas définie complètement.
- **Connaissances** : Ce sont des connaissances sur :
 - les agents AR du groupe (caractéristiques statiques des ressources).
 - la localisation des autres agents AF du système coopératif.
 - Contraintes qui maintiennent l'association de groupe (par exemple : ressource=BD Projets et département=informatique).

➤ **Agent négociateur**

Agent négociateur est un agent mobile chargé de s'occuper de la requête de client, il sera crée par AF en cas ou la requête est incomplète.

L'agent négociateur recopie la liste des caractéristiques des agents qu'il retrouve auprès du fédérateur, ensuite il migre vers le site du client et établit de façon interactive la requête de service complète, AF détruit l'agent négociateur.

II. La modélisation du système :

Nous avons choisi le langage AUML pour la modélisation, ce formalisme a comme but de recommander une technologie pour l'adoption d'une sémantique, d'une syntaxe abstraite communes pour l'analyse et la conception des méthodologies basées sur les agents.

Cette technologie devra permettre de couvrir le cycle de vie des produits et des outils de travail AUML et être en accord avec les spécifications existantes de FIPA et OMG (Object Management Group).

- La modélisation met en évidence quatre composantes de la modélisation largement utilisée dans pratiquement toutes les méthodologies :
 - Diagramme de cas d'utilisation.
 - Diagramme de séquence.
 - Diagramme de collaboration.
 - Diagramme de classes d'agents (remplace le diagramme de classes d'objets).
 - Diagramme d'activité
- Pour la modélisation du comportement de notre système nous utilisons le diagramme de cas d'utilisation (vue utilisateur).
- Pour la modélisation des interactions, échange de message et les protocoles entre les agents nous utilisons le diagramme de séquence (comportement du système).
- Pour spécifier de manière très précise l'ordre et les conditions d'envoi des messages entre les agents nous utilisons le diagramme de collaboration.

- Pour la modélisation de la vue de conception statique de notre système nous utilisons le diagramme de classes d'agents (structure du système).
- Pour la modélisation de la vue de conception dynamique de notre système nous utilisons le diagramme d'activités d'agents.

II.1 Diagramme de cas d'utilisation :

Un diagramme de cas d'utilisation représente un ensemble de cas d'utilisation et d'acteurs ainsi que les relations. On applique ces diagrammes pour illustrer la vue de cas d'utilisation statique d'un système, ils sont particulièrement importants pour l'organisation des comportements d'un système.

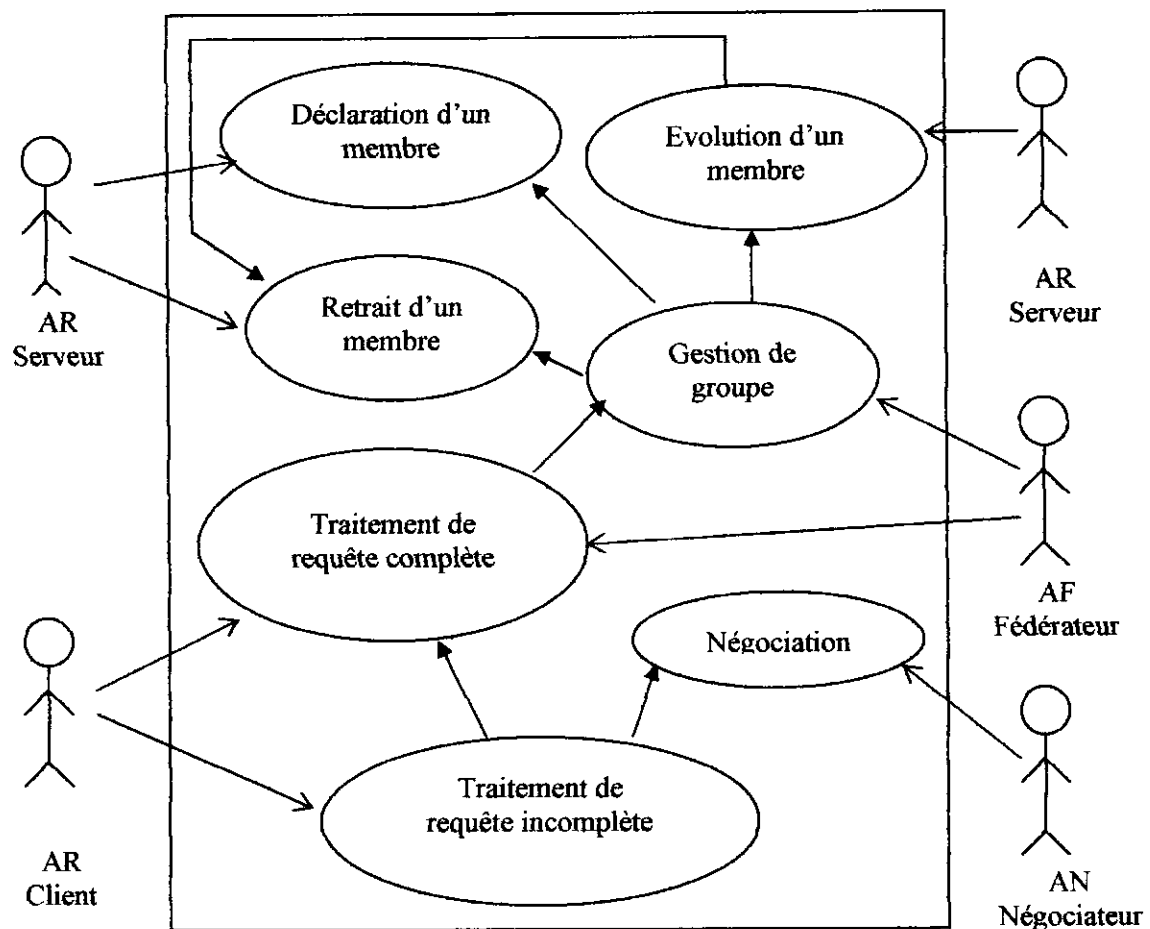


Figure 2.3 : Diagramme de cas d'utilisation

Legende :

- : Utilise
- : Déclenche

Nous avons huit cas d'utilisation dans notre système :

1. résolution de la requête
2. traitement de la requête complète
3. traitement de la requête incomplète
4. négociation
5. gestion de groupe
6. évolution d'un membre
7. déclaration d'un membre
8. retrait d'un membre

➤ **Les différents rôles d'un agent :**

- AR client lance la requête.
- AF commence le traitement de la requête (le découpage de la requête, la gestion de groupe, le choix du groupe le plus adapté, la création de l'AN dans le cas d'une requête incomplète, construction de la requête globale.....etc).
- Dans le cas d'une requête incomplète AF traite la requête en créant un agent négociateur AN responsable de la négociation des caractéristiques de la requête et l'établissement de la requête complète.
- AR serveur responsable de la résolution de la requête.
- AR se déclare au moment de sa création, son intention de se retirer et les changements de ses caractéristiques.

Nous pouvons ajouter une illustration des rôles joués par les agents du système.

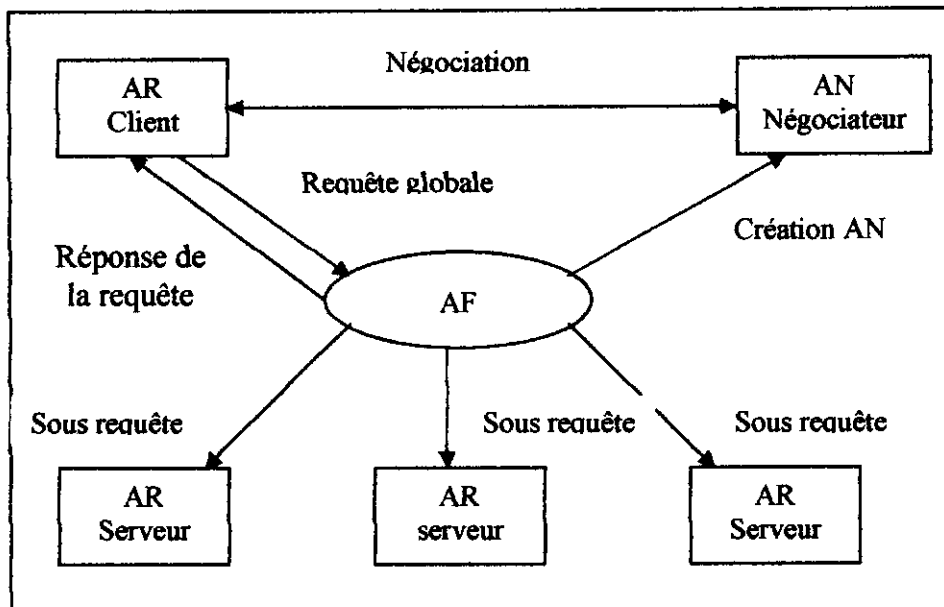


Figure 2.4 : Les principaux rôles joués par les agents du système

II.1.1 Organisation des interactions dans les SMA :

Dans notre système la coopération entre les agents est basée sur des messages de requête de service, l'organisation des interactions met en évidence 5 scénarios essentiels:

1) Scénario de la résolution de la requête complète :

1. AR client demande de l'agent fédérateur AF la résolution de la requête.
2. AF vérifie la requête.
3. AF envoie la requête aux AR serveurs du groupe.
4. A la réception de la requête, chaque AR serveur élabore une offre en comparant les éléments de la requête avec l'ensemble de ses compétences et envoie sa réponse à l'AF.
5. AF évalue les réponses et établit des contrats avec AR serveur dont les réponses contiennent des éléments déterminants dans la résolution de la requête.

6. AF procède au découpage de la requête en sous requêtes et envoie à chaque AR serveur contractant la sous requête qui correspond à ses compétences.
7. Les AR serveur résolvent les sous requêtes et renvoie les résultats au AF.
8. AF construit la réponse et la renvoie à l'AR client.

2) Scénario de la résolution de la requête incomplète :

1. AR client demande de l'AF la résolution de la requête.
2. AF vérifie la requête.
3. AF crée un agent négociateur AN ce dernier recopie la liste des caractéristiques des AR serveur qu'il retrouve auprès du AF.
4. AN migre vers le site du client pour accomplir la requête par la négociation.
5. AN établit la requête complète et l'envoie au AF.
6. AF détruit AN.
7. Reprendre l'étape 3 de la résolution de la requête complète.

3) Scénario de la déclaration d'un nouveau membre :

1. chaque AR se déclare au moment de sa création aux AF du système en signalant l'ensemble de ses caractéristiques.
2. chaque AF du système fait la comparaison entre ces caractéristiques et celles du groupe.
3. si elles sont compatibles AF le fait appartenir au groupe, sinon il rejette.
4. AF accepte AR dans le groupe, et met à jour ces accointances.

4) Scénario du retrait d'un membre :

1. AR signale au AF son intention de se retirer du groupe.
2. AF retire AR, et met à jour ces accointances.
3. AR met à jour ces accointances.

5) Scénario de l'évolution d'un agent :

1. l'agent AR ayant évolué signale le changement de certains de ses caractéristiques à tous les fédérateurs AF.
2. après que chaque AF vérifie les caractéristiques, l'AR peut être ajouter, retirer ou garder dans le groupe.



II.1.2 protocole d'allocation de ressources :

Pour le filtrage et la sélection des AR serveurs qui sont compatibles avec la requête, nous avons utilisé le réseau contractuel, c'est un mécanisme d'allocation de tâche fondé sur la notion d'appel d'offre.

Nous avons choisi cette technique car elle est simple à comprendre et à utiliser, et qui donne une vision plus naturelle à la coopération de ressources.

L'appel d'offre s'effectue en quatre étapes, Nous allons les appliquer pour la sélection des AR serveur qui sont capables de résoudre la requête, chaque étape est suivie par le message qui lui correspond :

1. à la réception de la requête R, AF envoie une description de celle la à tous ceux qu'il estime pouvoir répondre
→ *Appel Offre (R, AF)*.
2. Les AR serveur (offrants) élaborent une proposition P qu'ils envoient à l'AF :
→ *proposition (R, P) est une réponse positive de AR serveur*
→ *PasIntéressé (R, S) est une réponse négative de AR serveur.*
3. AF reçoit et évalue les propositions et attribue le marché au meilleur offrant.
4. AR serveur qui a reçu le marché et qui devient ainsi le contractant envoie un message à l'AF lui indique qu'il est toujours d'accord pour accomplir la requête et qu'il s'engage de ce fait à la réaliser.
→ *attribuer (R, C, AF) est le message de AF à AR serveur pour lui signifier qu'il lui attribué le contrat C portant sur la requête R.*

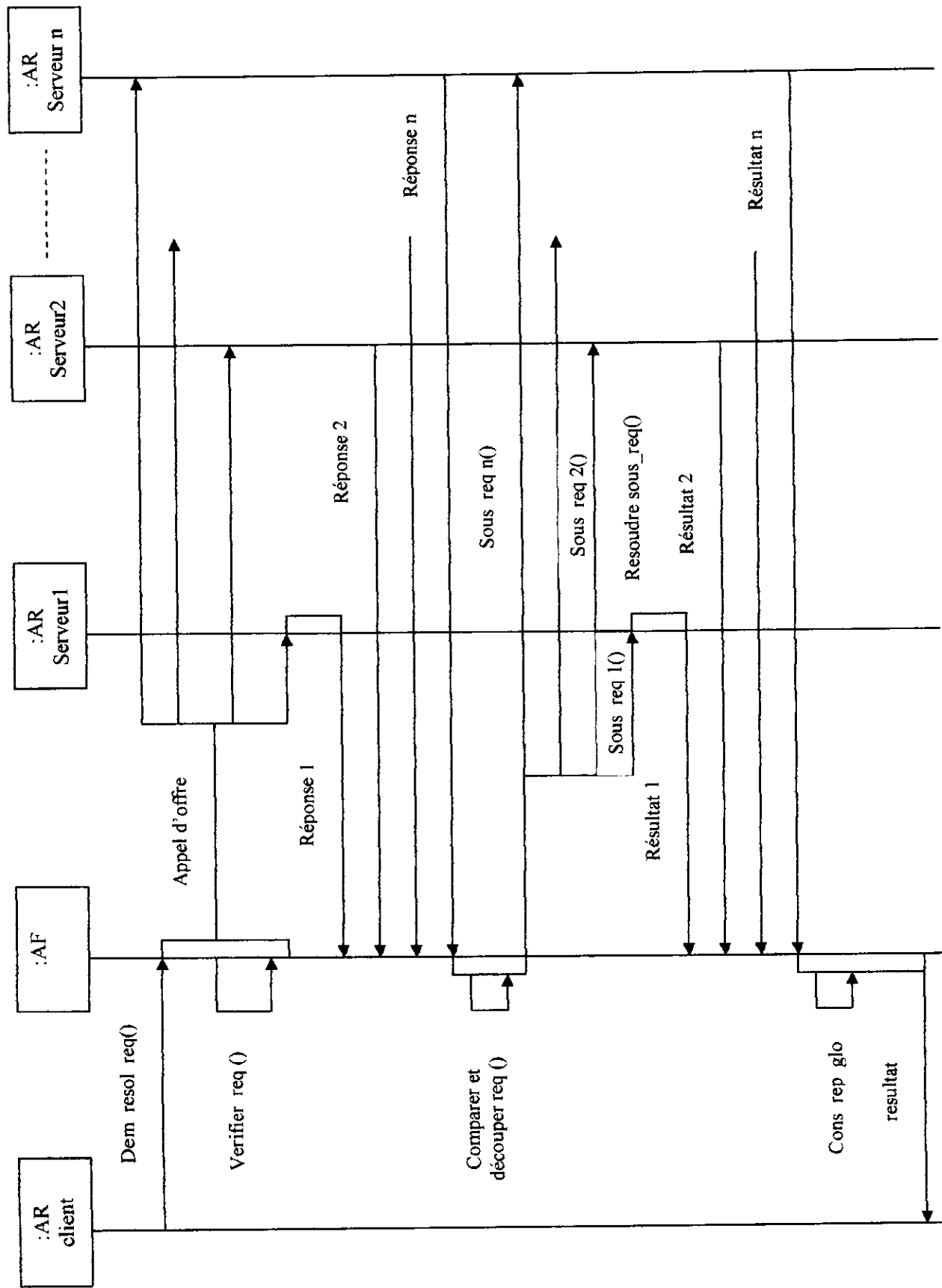
II.2 Diagramme de séquence

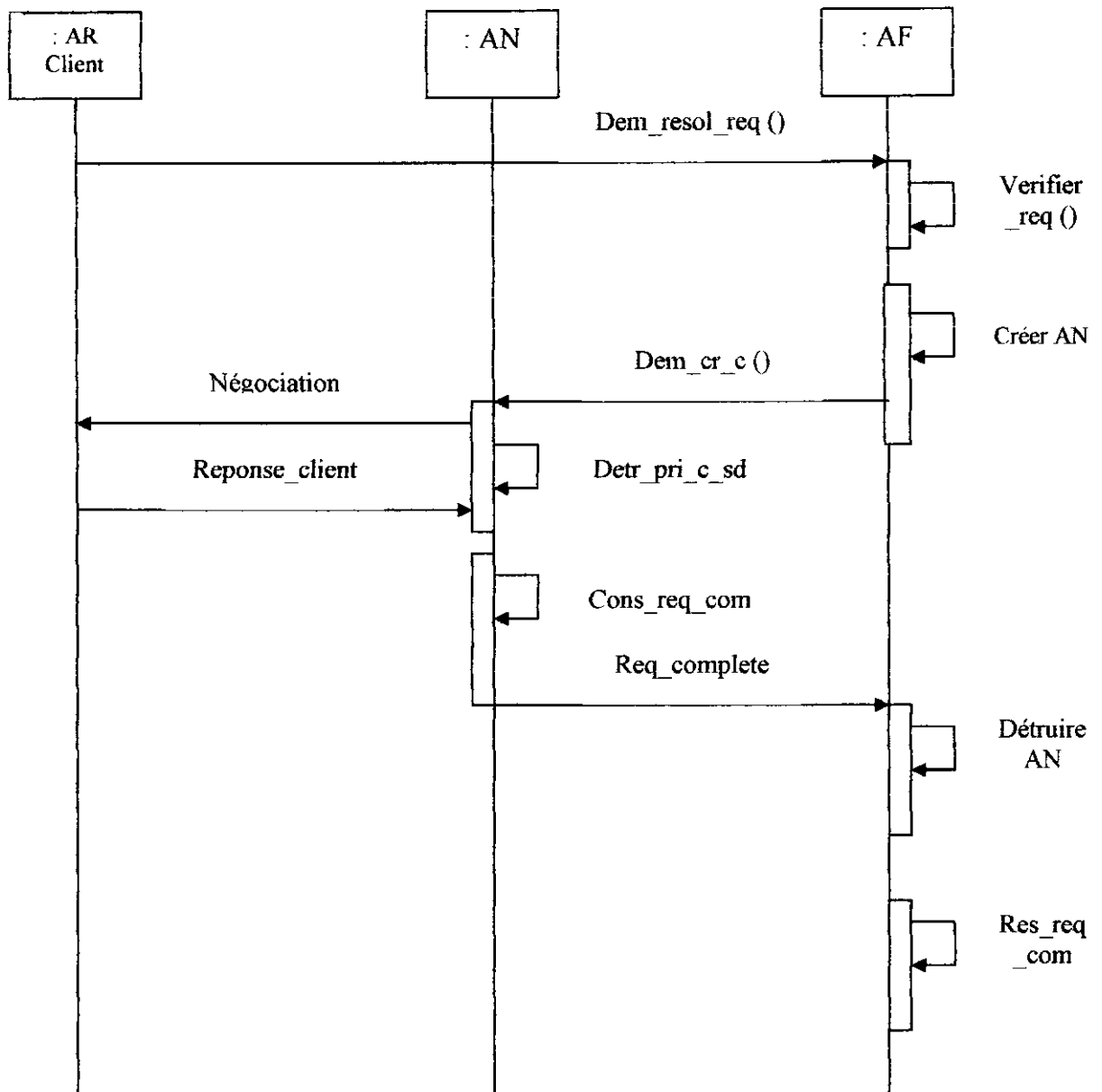
Les diagrammes AUML sont des adaptations au contexte des systèmes multi agents des diagrammes UML, le diagramme de séquence AUML est très semblable au diagramme de séquence UML. Des opérateurs spécifiques ont été ajoutés, il permet de décrire un protocole d'interaction mettant en jeux divers agents de différents rôles.

Nous allons associer pour chaque scénario un diagramme de séquence

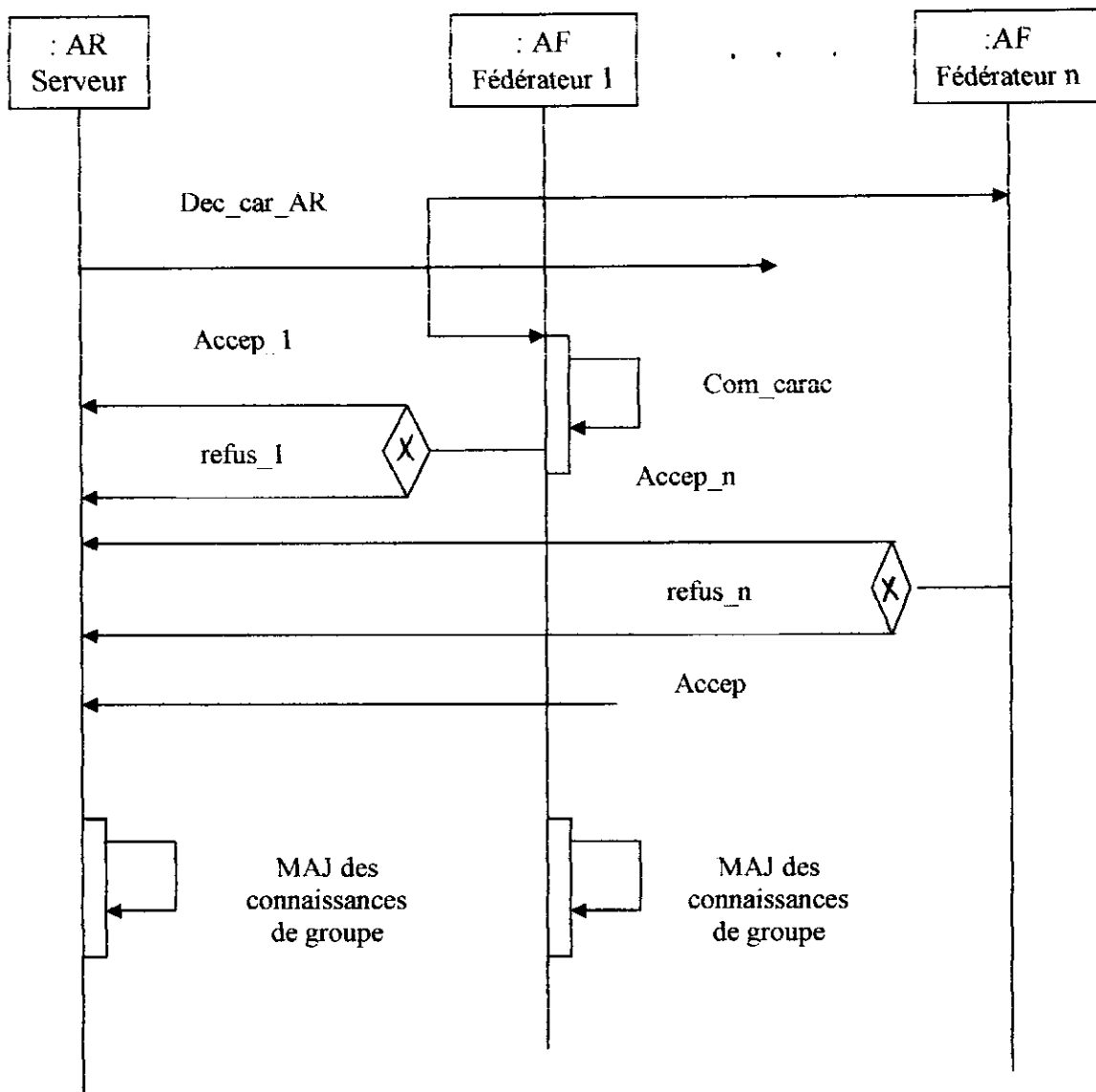
Légende :

Message	signification
Dem_res_req ()	demande résolution de la requête
Dem_cr_c ()	demande de construction de la requête complète
Rec_c_ARs ()	récupération des caractéristiques des AR serveur
Detr_pri_c_sd	détermination des principales contraintes du service demandé
Dec_car_AR	déclaration des caractéristiques de l'agent ressource
Accep	acceptation de l'agent
Sin_ret	signalisation de l'intention de se retirer
Acc_rec	accuser de réception
Cons_rep_glo	construction de la réponse globale
Sous_req	sous requête
Res_req_com	résolution de la requête complète
Res_sous_req	résolution des sous requêtes
Cons_req_com	construire requête complète
Com_carac	comparaison des caractéristiques
Dec_n_c	déclaration des nouvelles caractéristiques
Mem_ret_gp	membre retirer du groupe
Mem_aj_gp	membre ajouter au groupe
Mem_gar_gp	membre garder dans le groupe
Ap_off	appel d'offre
Res_gl	résultat global
Req_comp	requête complète
Negocier_car_ser	négozier les caractéristiques du service
Fil_ser_adap	filtrage des serveurs les plus adapté
Sig_ens_car	signalisation de l'ensemble de caractéristiques
V_req	vérifier la requête

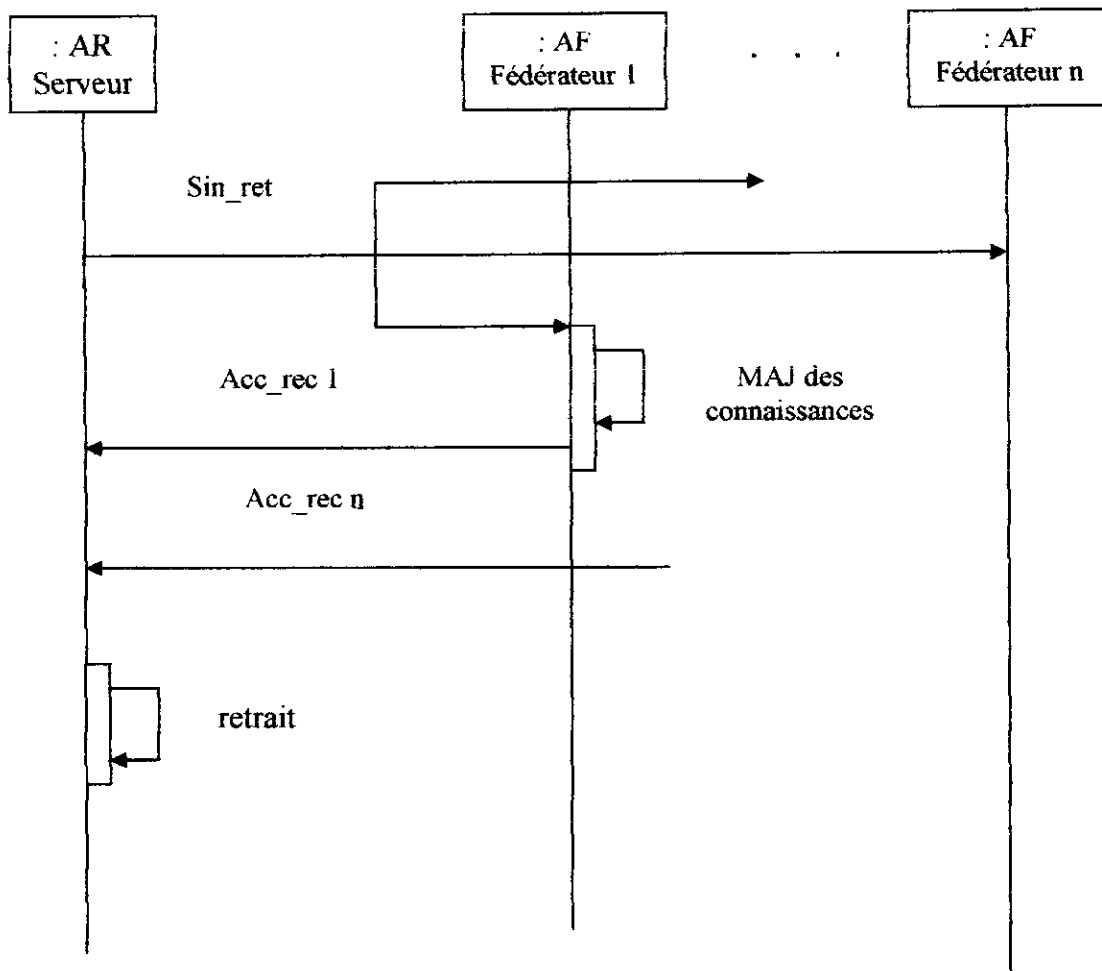




2. La résolution d'une requête incomplète



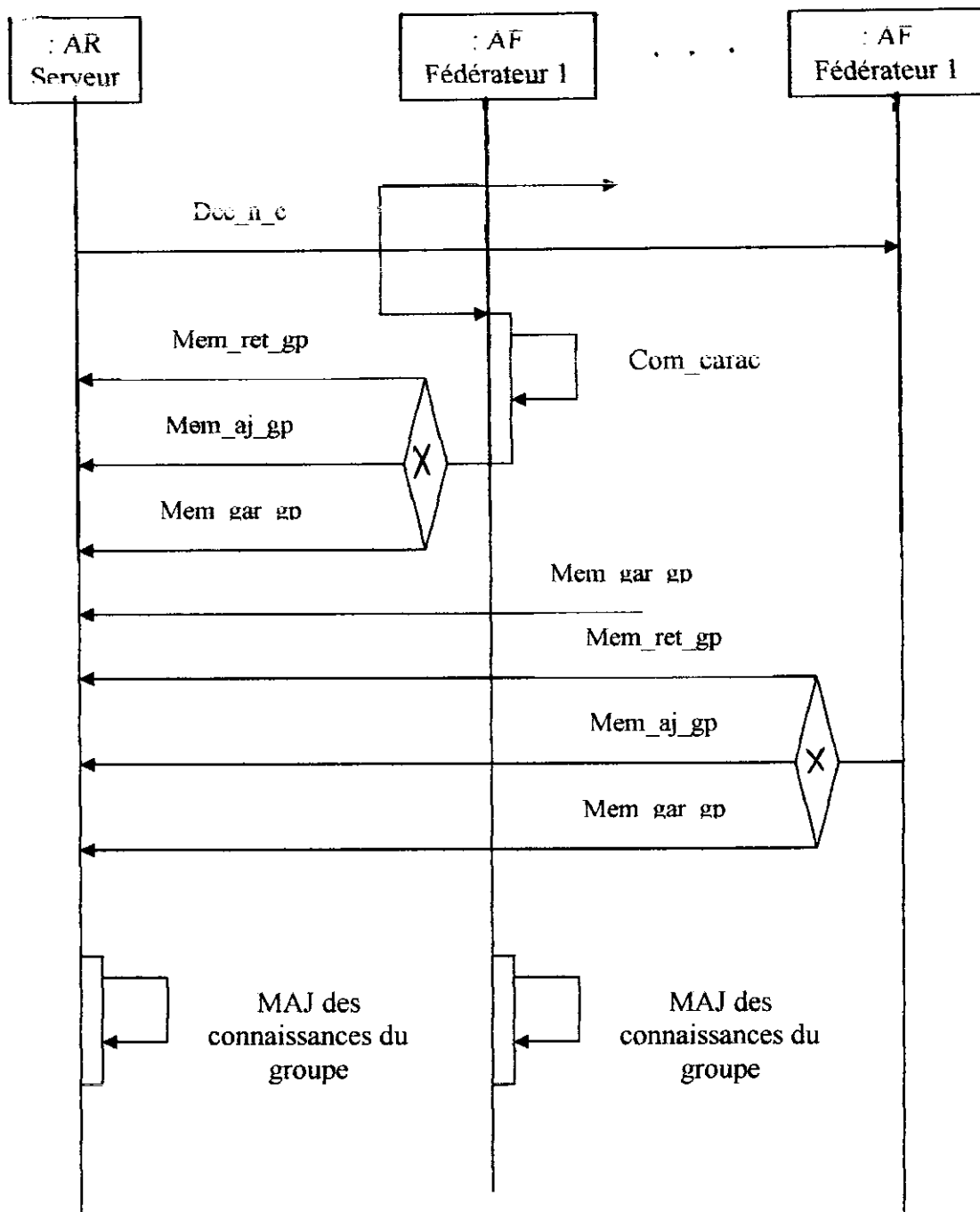
3. L'ajout d'un agent



4. Le retrait d'un agent

Remarque

L'agent ressource AR signale son intention de se retirer aux AF qu'il appartient



5. Evolution d'un agent

11.3 Diagramme d'interactions de classes d'agents :

Le diagramme de classe d'agent permet de capturer les agents types et les interactions entre les agents de notre système.

AUML permet de représenter plusieurs niveaux d'abstraction lors de la conception des diagrammes de classes

Deux niveaux d'abstractions sont couramment utilisés le niveau conceptuel et le niveau d'implémentation

1. le niveau conceptuel : est une vue assez haute du SMA éliminant toute les informations qui sont superficielles pour comprendre la structure du système : les attributs, les opérations, les rôles,.....
2. le niveau implémentation : est une vue plus détaillée sur le contenu des agents

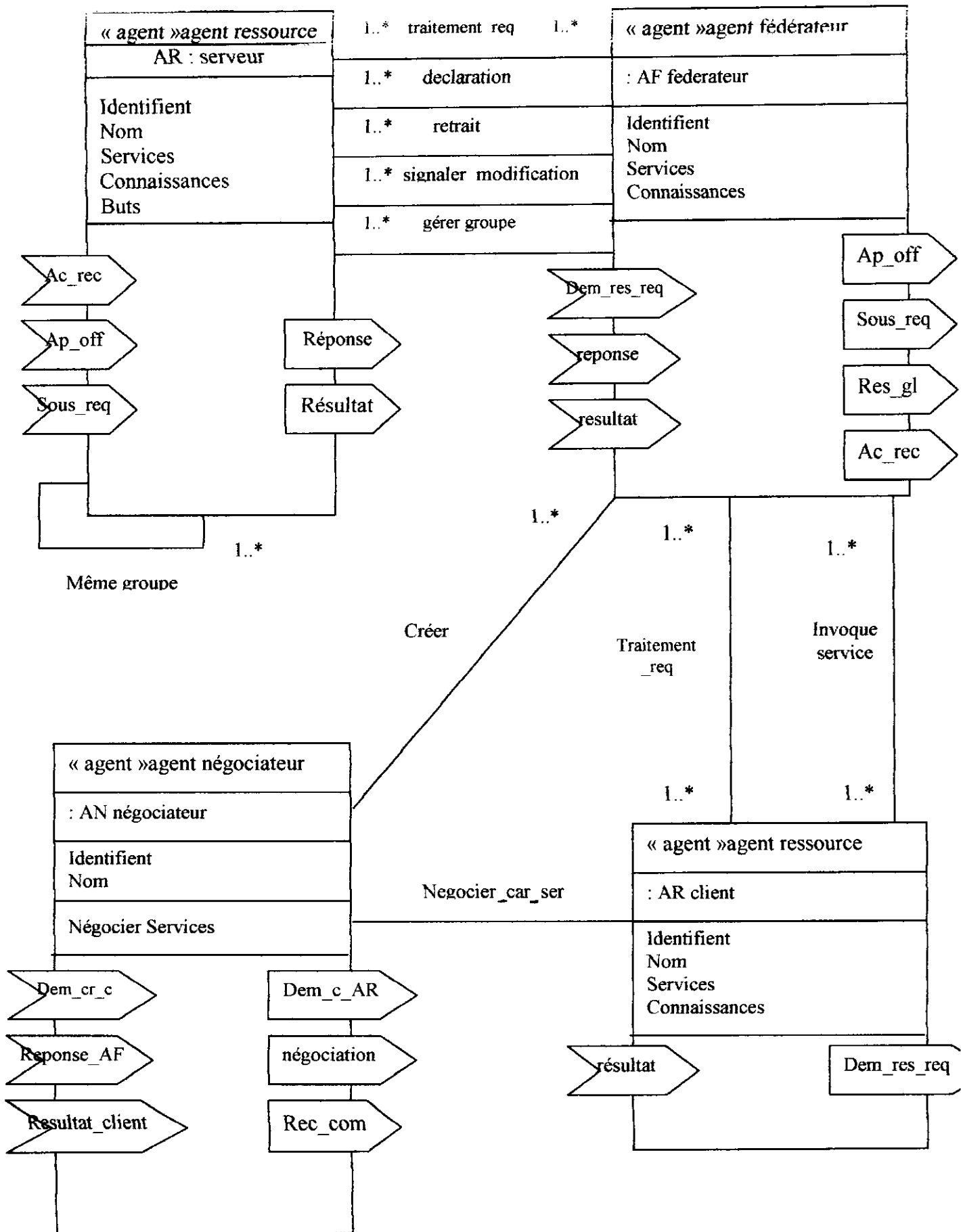


Figure 2.5 : Diagramme d'interactions de classe d'agents

2.1 Diagramme d'activité :


Le diagramme d'activité fait partie des cinq diagrammes utilisés pour la modélisation des aspects dynamiques des systèmes :

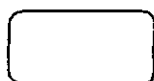
Un diagramme d'activités est principalement un organigramme qui montre le flot de contrôle d'une activité à l'autre

1. Le diagramme d'activité d'un agent :


Un diagramme d'activité d'un agent est généralement représenté comme suit :


Légende :

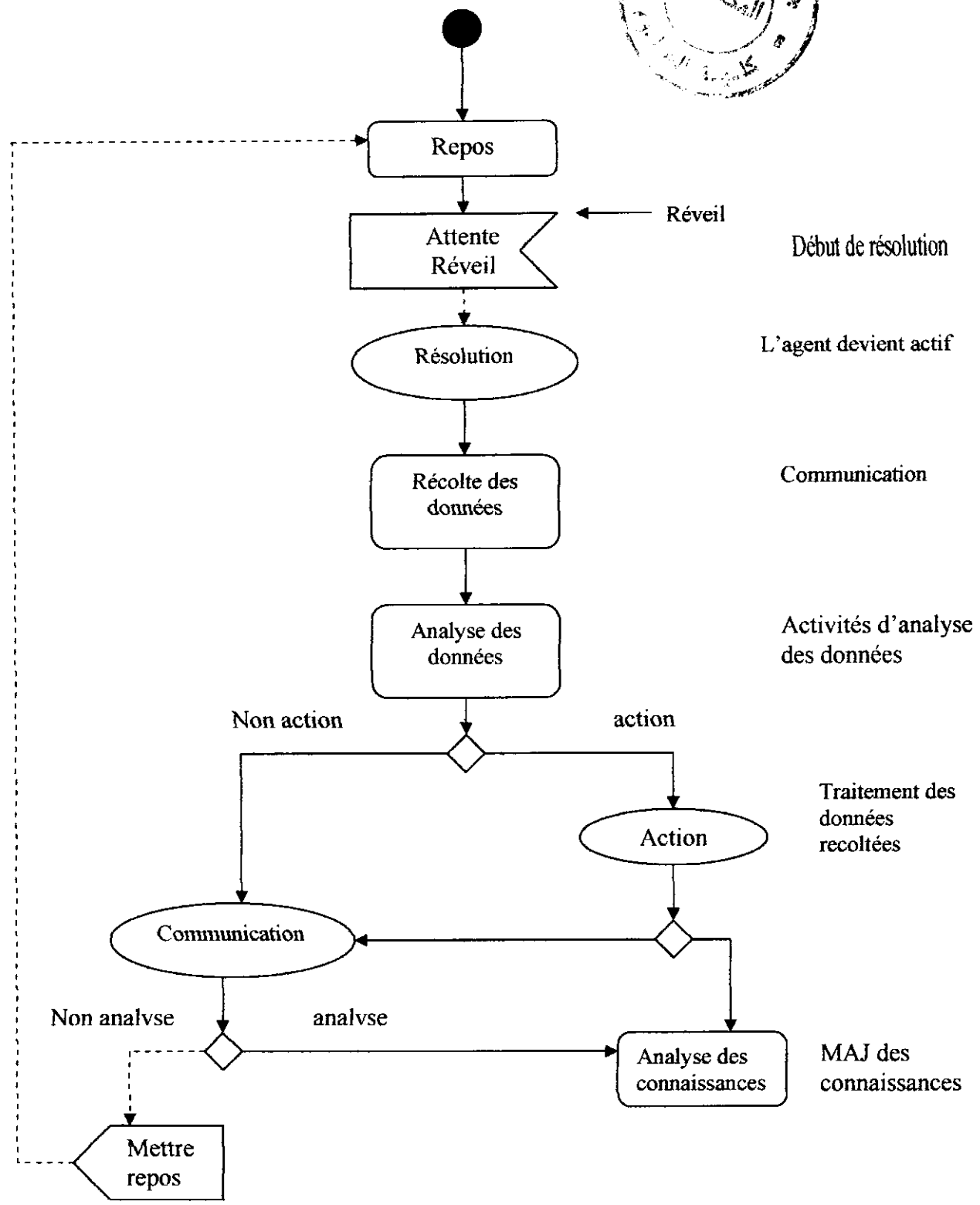
 : État initial

 : État d'agent

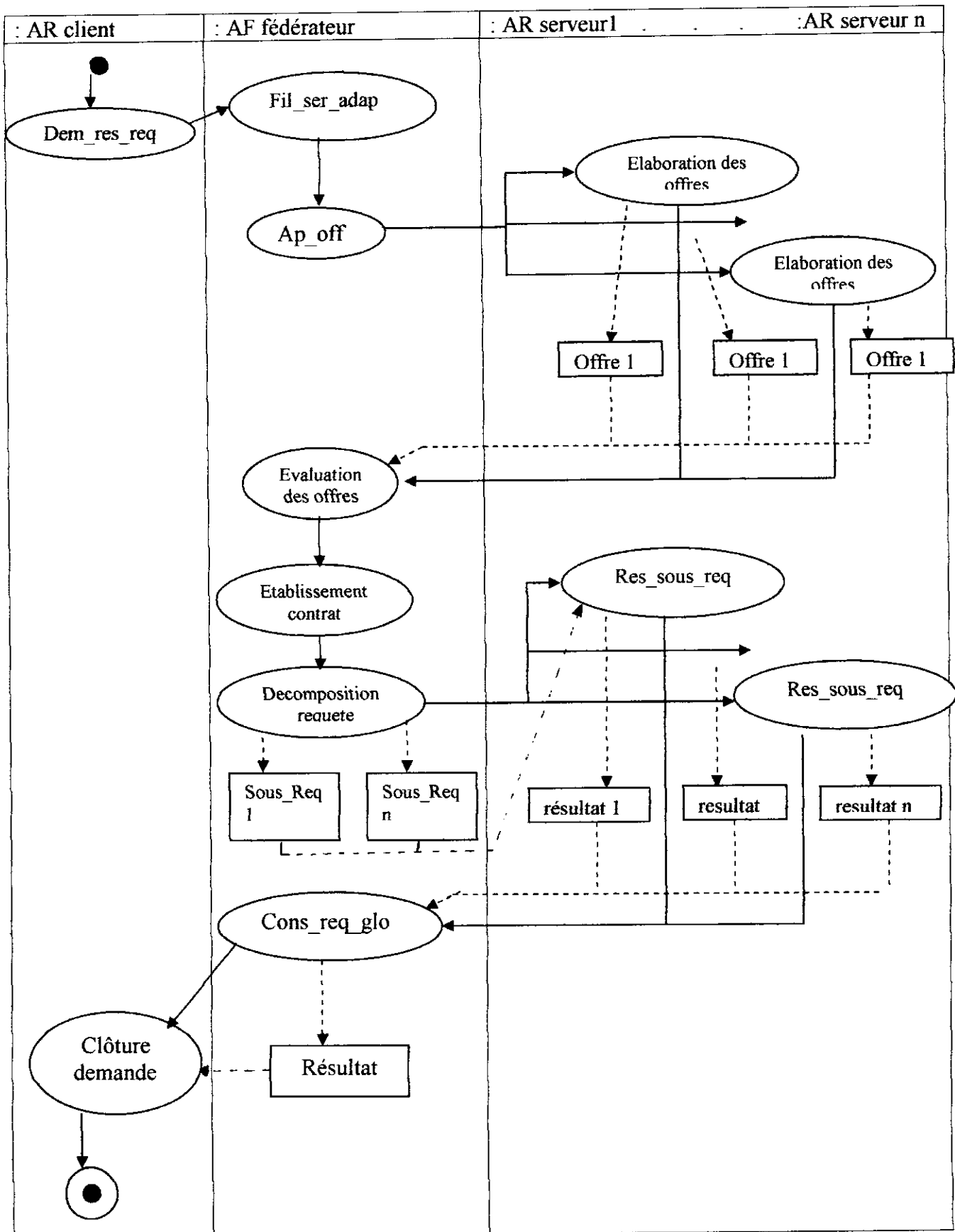
 : Action

 : Ou inclusif

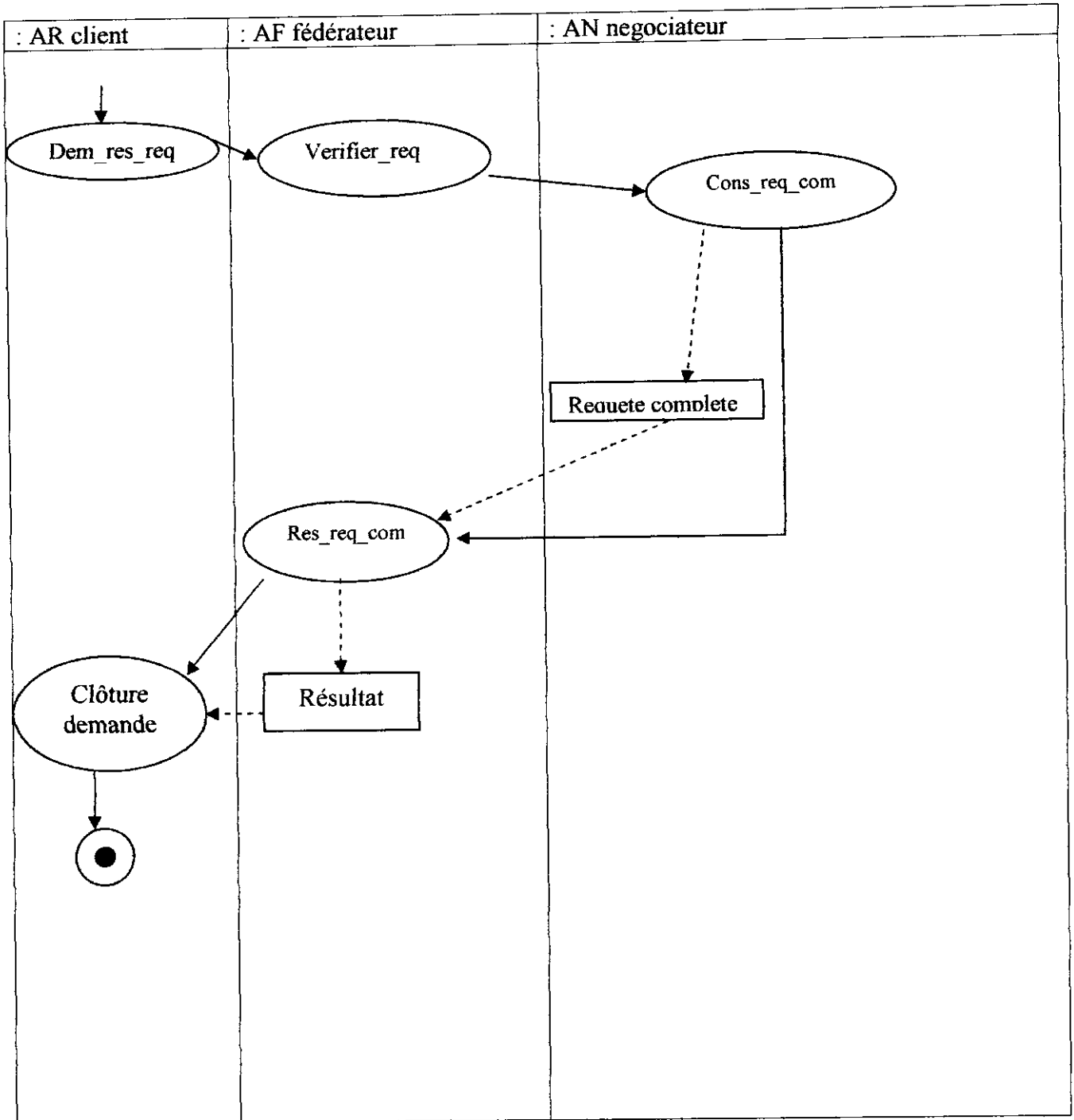
 : État final



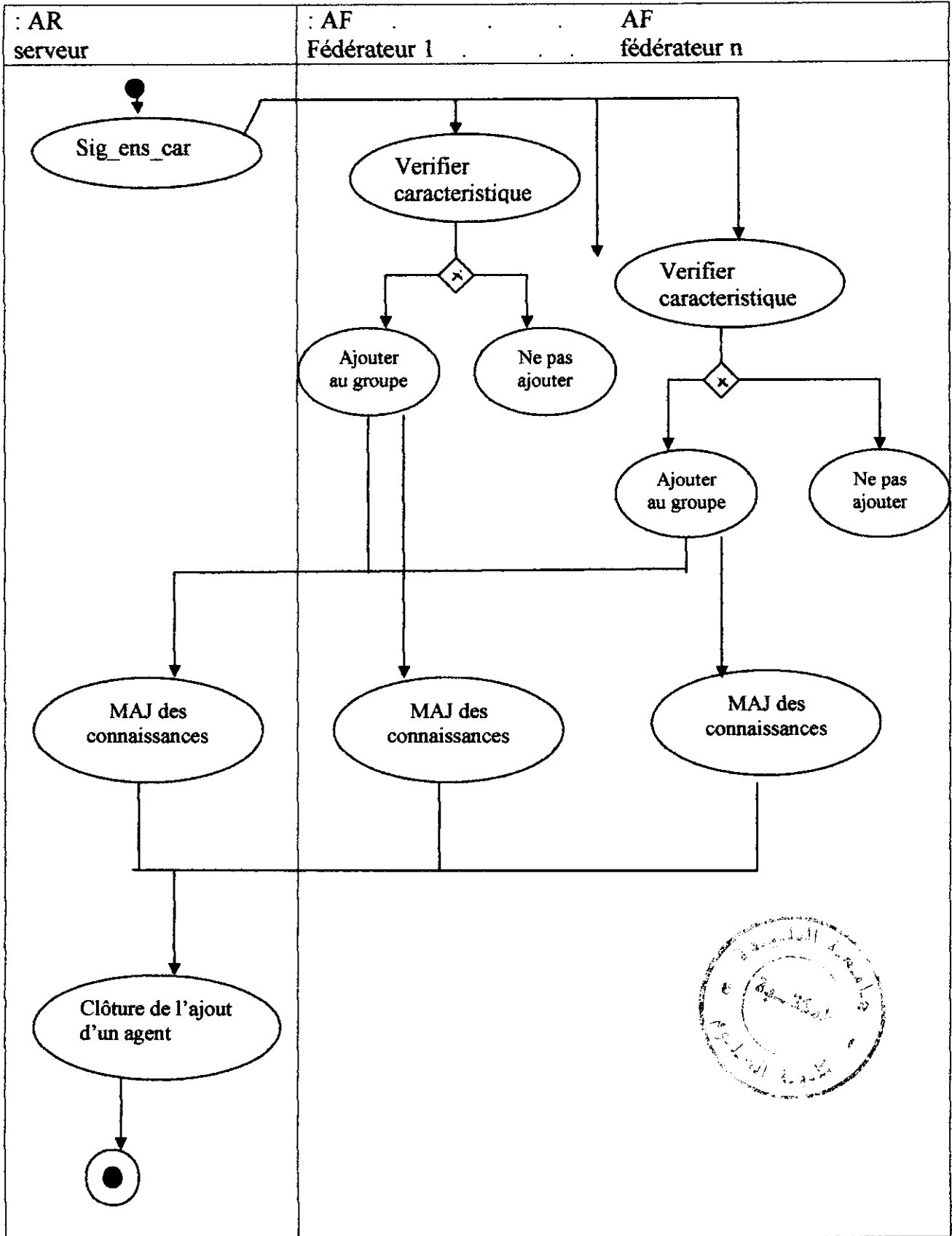
1. diagramme d'activité d'un agent



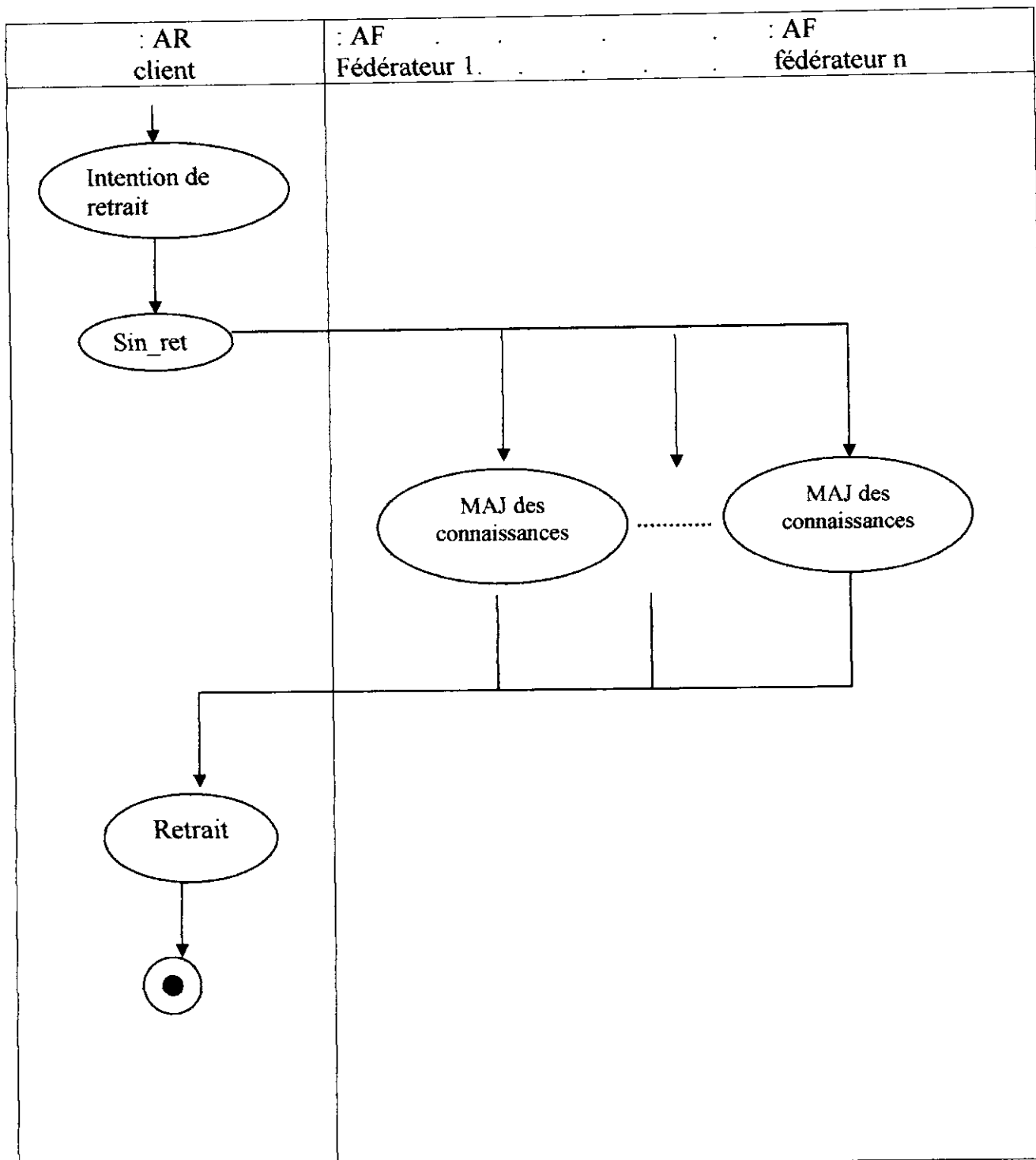
2. La résolution de la requête complète



3. La résolution de la requête incomplète



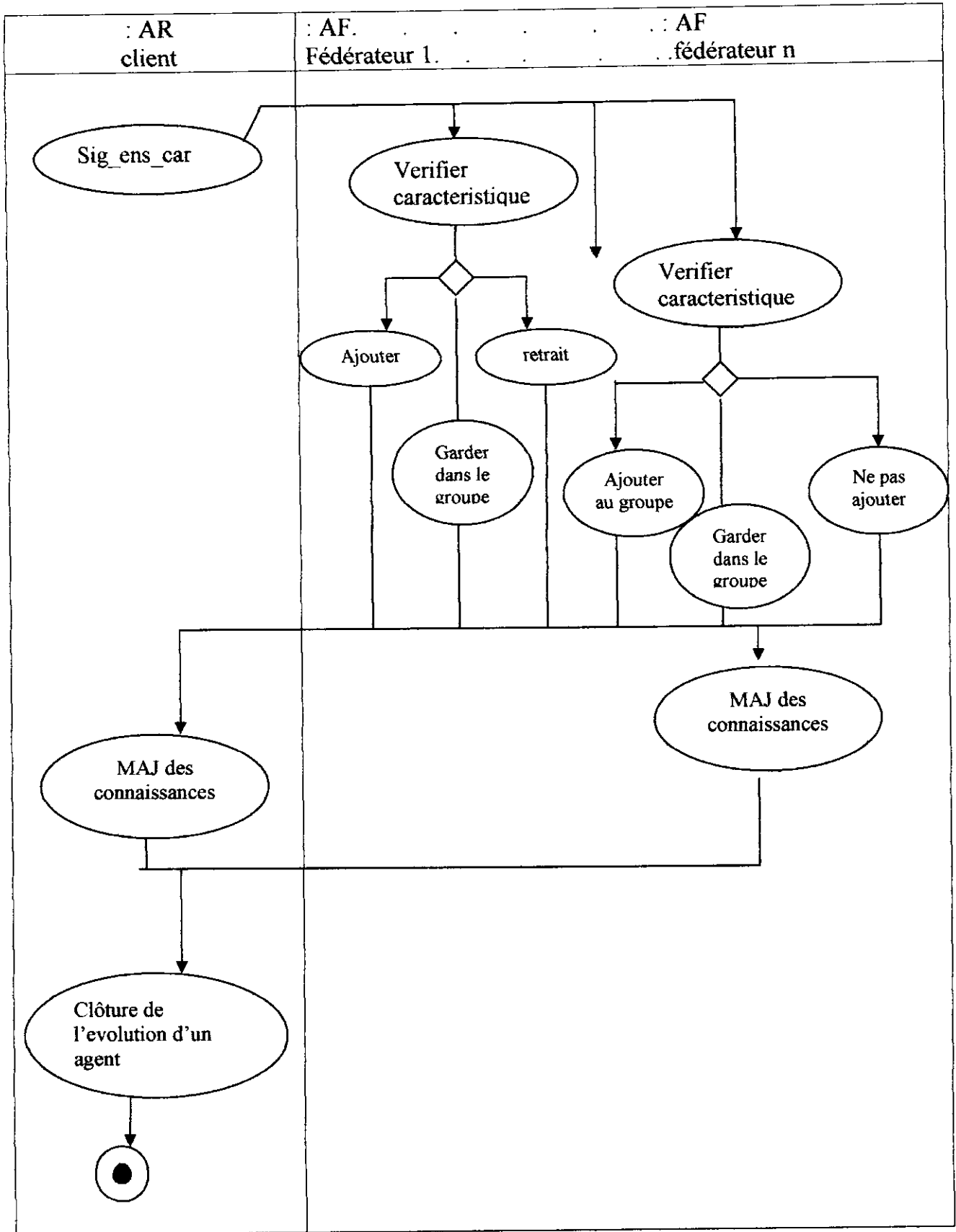
4. L'ajout d'un agent



5. Le retrait d'un agent

Remarque :

AR signale son intention de se retirer seulement au AF qu'il fait appartenir.



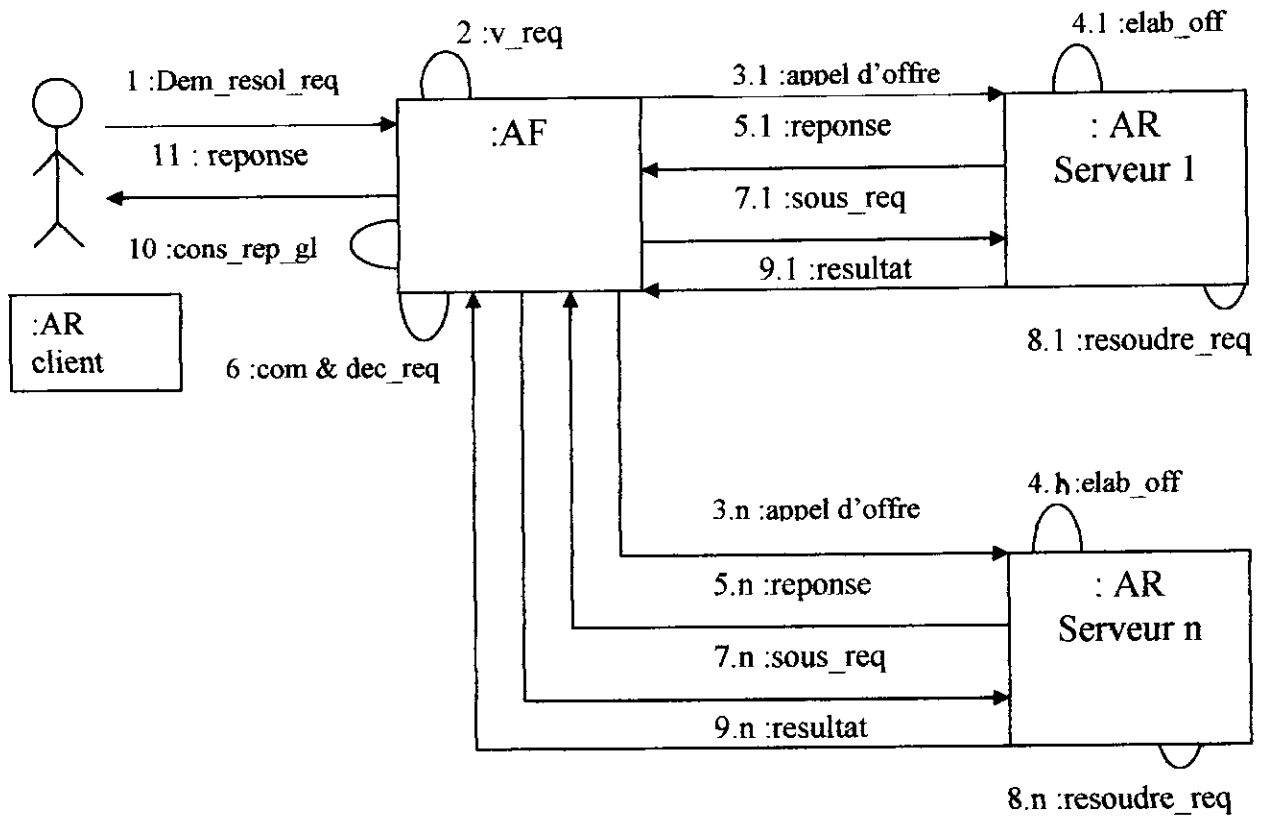
6. Evolution d'un agent

II.5 Diagramme de collaboration :

Un diagramme de collaboration met en évidence l'organisation des agents qui participent à une interaction

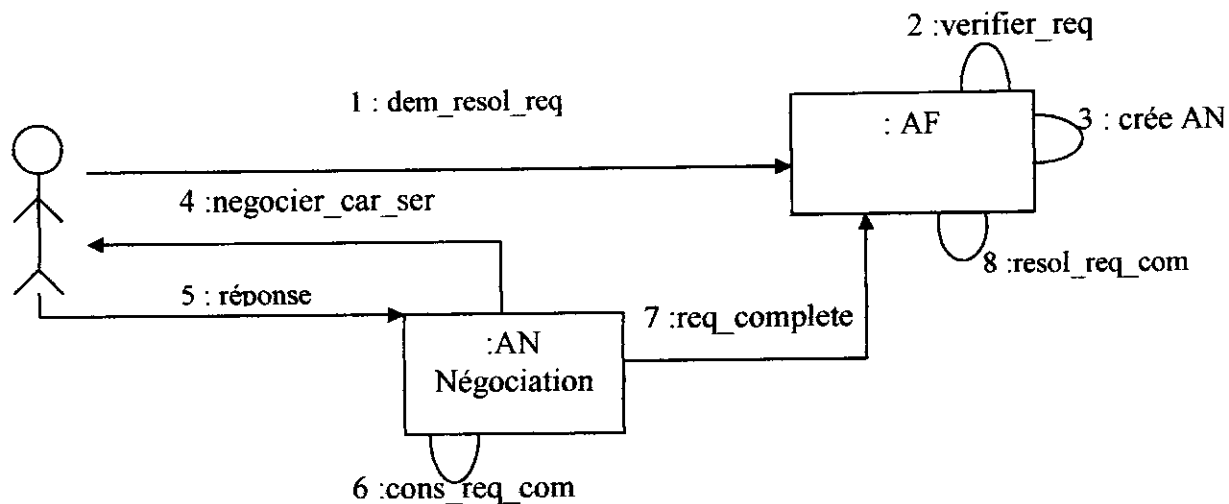
Deux caractéristiques distinguent les diagrammes de collaboration des diagrammes de séquence :

1. le chemin pour indiquer comment un agent est lié à un autre
2. le numéro de séquence pour indiquer le classement chronologique d'un message.

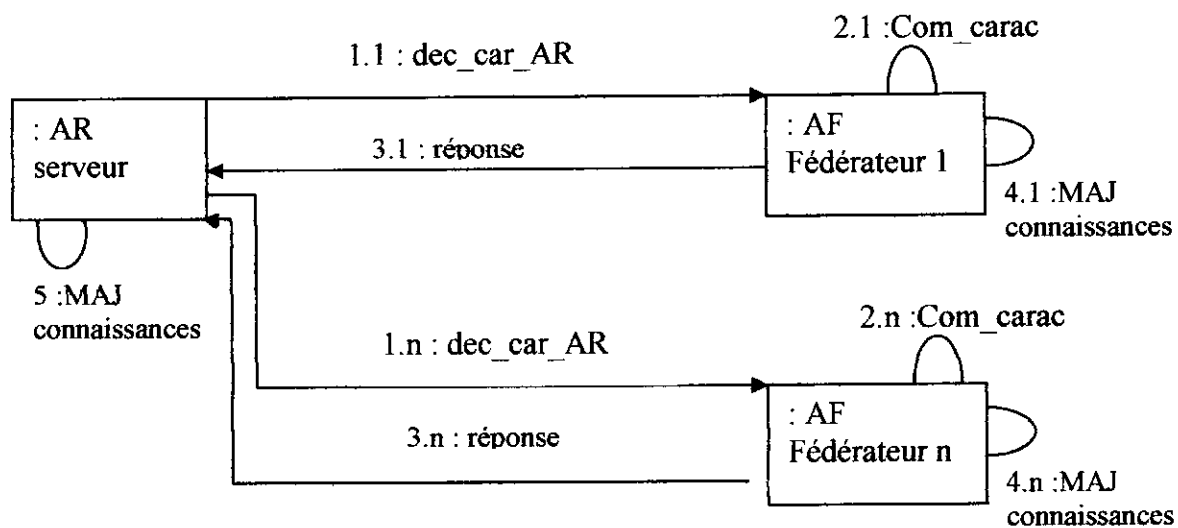


1. La résolution de la requête complète

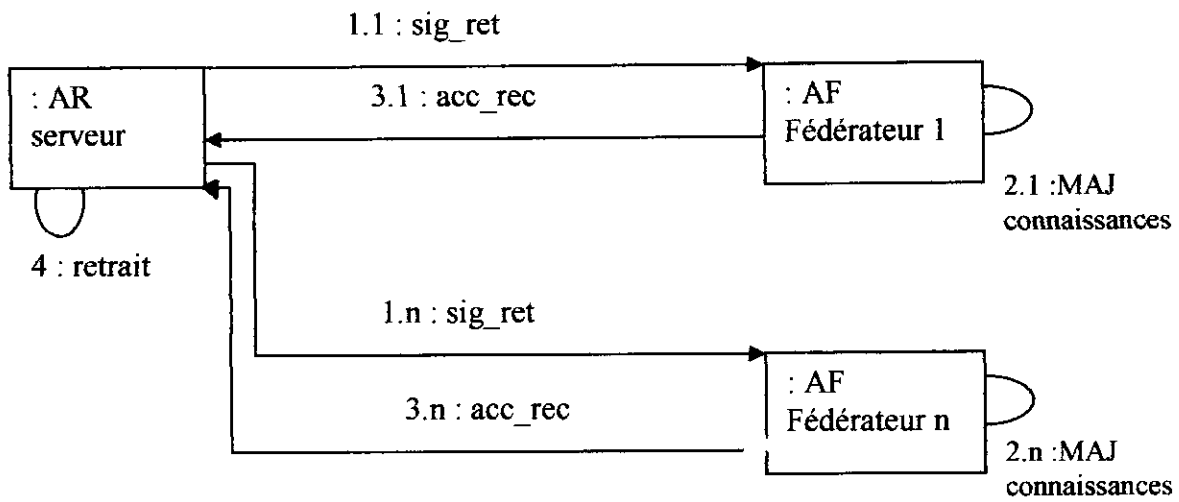




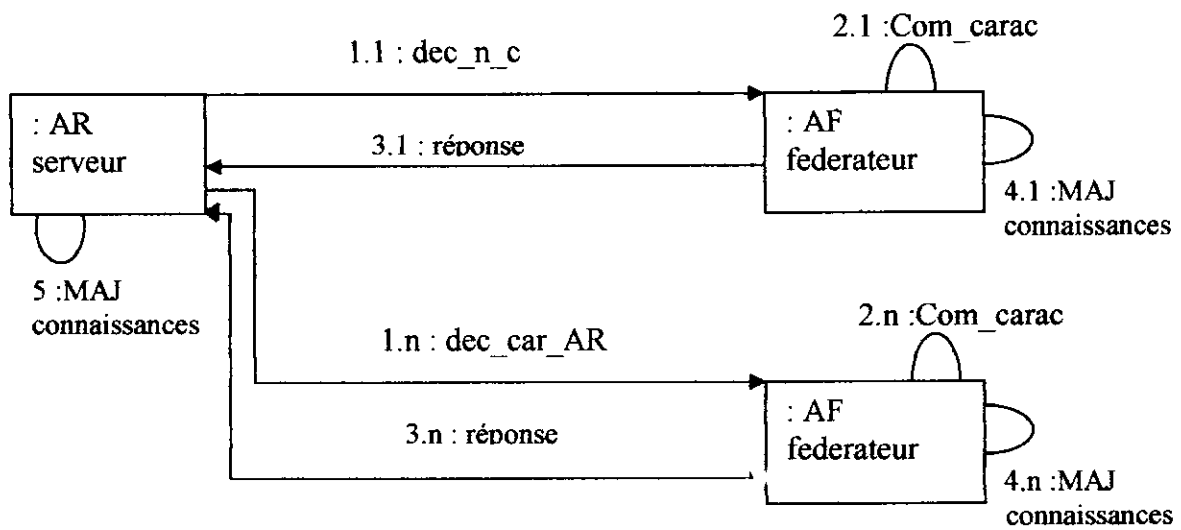
2. Résolution de la requête incomplète



3. L'ajout d'un agent



4. Le retrait d'un agent



5. Evolution d'un agent

III. Etude de cas : Système de réservation de salle multi-agents

L'objectif de cette partie est de mettre en œuvre la modélisation multi-agents dans un cas concret et simple à savoir un système de réservation de salles multi-agents.

L'outil développé est un prototype pédagogique de simulation afin d'illustrer le fonctionnement et les interactions entre les agents du SMA qui est au service de la coopération.

A travers cette étude de cas, nous voulons mettre en évidence une base opérationnelle multi-agents permettant l'intégration de différentes ressources d'un SI distribué.

Une vision très simple d'une requête globale du système est donnée par le schéma externe suivant :

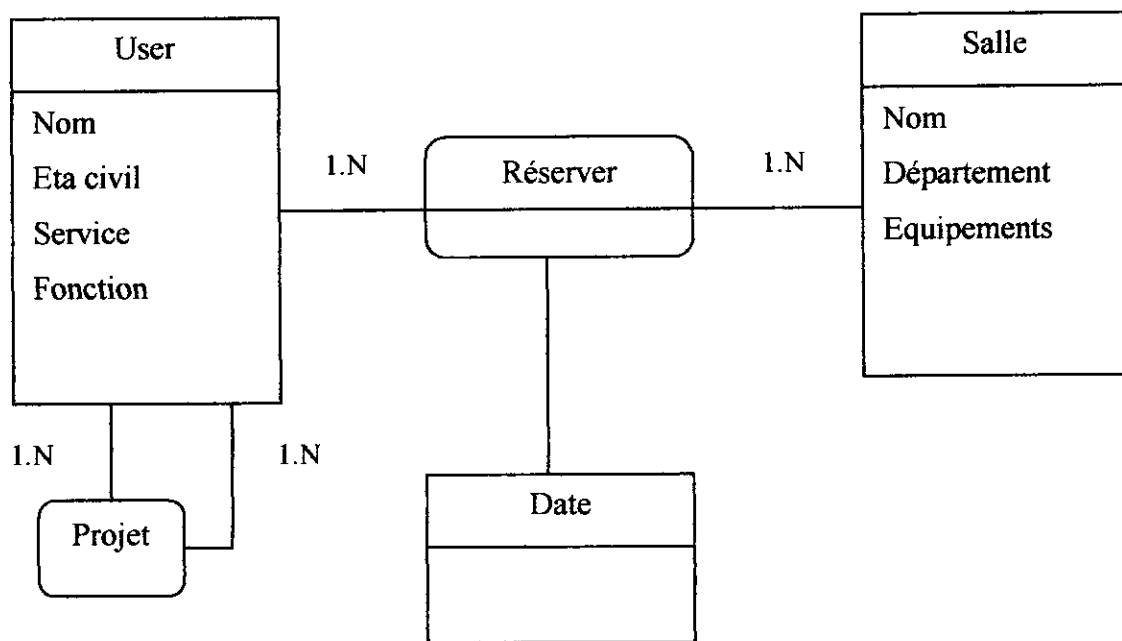


Figure 2.6 : schéma externe

Ce schéma ne représente que le sens des requêtes globales. Il sera repris dans le contexte du SMA à travers le diagramme d'interaction de classes d'agents et le diagramme de collaboration pour la prise en compte de la coopération.

III.1. Description du système :

Les ressources impliquées dans le système sont les salles et les utilisateurs. Le système peut être étendu pour intégrer d'autres ressources, matérielles par exemple (imprimantes, scanners, documents,.....), chacune de ces ressources est représentée de façon exclusive par un agent, celui-ci joue un rôle d'interface complète entre la ressource et le reste du système (les autres ressources).

Ainsi, lorsqu'un utilisateur désire réserver une salle, ou annuler une réservation, il a accès au service correspondant via son agent *user* qui se charge de contacter les agents *salles*. l'agent propose également un ensemble de services locaux dans l'esprit de maintenir l'autonomie des ressources. Pour pouvoir représenter une ressource, les agents en un rôle de base de données locale.

En effet, les agents maintiennent un ensemble d'information sur la ressource qu'ils représentent.

Les agents ressources du système sont regroupés en groupes dont chacun est fédéré par un agent fédérateur.

Afin de pouvoir développer ce prototype nous avons opté pour un ensemble de domaines. Nous citons par exemple :

- G1 : regroupe toutes les ressources sauf les salles.
- G2 : regroupe tout les utilisateurs du département d'informatique.
- G3 : regroupe toutes les ressources du département d'informatique.
- G4 : regroupe toutes les salles sauf celles situées au département d'informatique.
- G5 : regroupe les salles disposant de plus de 1000 chaises.

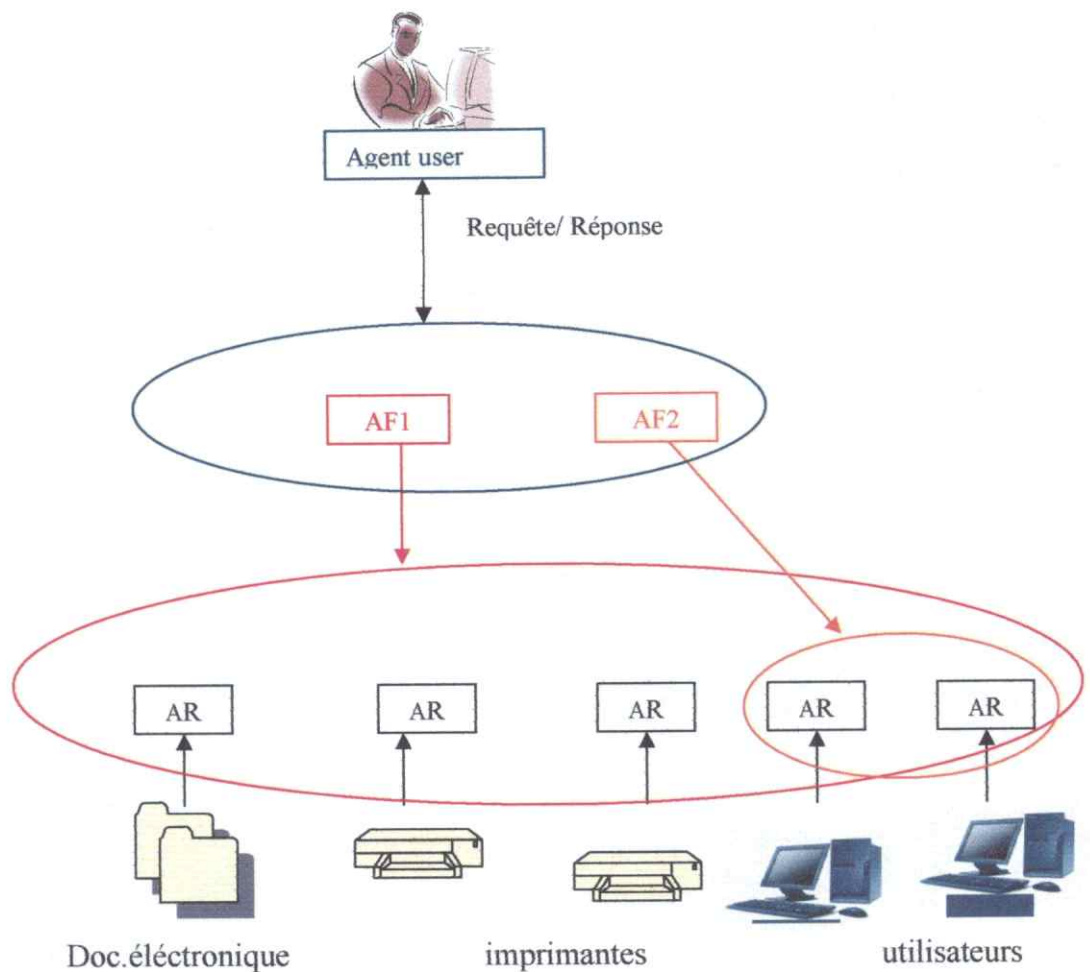


Figure 2.7 : une vue globale du SMA

La coopération entre les agents est basée sur des messages de requêtes de service, citons à titre d'exemple :

- Réserver salle.
- Annuler réservation.
- Utiliser ressource : imprimante,...
- Communiquer -user- entre utilisateurs

Un exemple de l'organisation de groupes ainsi que celle des interactions est illustré dans la figure 2.7

2. description des agents et des services :

2.1. les agents ressource *AR* :

Les agents *salle* et les agents utilisateurs *user*, représentent les ressources du système. Ils possèdent un caractère permanent, lié à la présence de ces ressources, les agents ont une représentation interne de leur ressource à travers des caractéristiques.

L'agent *salle* :

- Connaît l'emplacement, le mobilier (nombre de table, nombre de chaises, dispositif de projection).
- Tient à jour le planning de réservations.

L'agent *user* :

- Connaît l'état civil, le service, la fonction,...
- Maintient un historique des réservations qu'il a effectuées.

2.2. les agents fédérateurs *AF* :

Ce sont les agents qui fédèrent les groupes. Chaque domaine d'intérêt est fédéré par un AF qui joue le rôle d'intermédiaire entre les ressources et les utilisateurs, il coordonne l'ensemble des activités des agents d'un même groupe.

Chaque AF dispose des caractéristiques des agents ressources qu'il a dans son groupe. Notons ici que la construction de groupe se fait de manière dynamique, ce qui implique que, dans le temps, la définition du groupe peut changer.

2.3. les agents auxiliaires :

le système met également en jeu deux agents auxiliaires :

2.3.1. Agent Page Jaune *PJ* :

C'est un agent lié à l'exécution du protocole de coopération basé sur la gestion de groupes. L'agent PJ regroupe toutes les caractéristiques de domaines liés aux AF. A l'invocation d'une requête de service (réservation par exemple), les caractéristiques de la requête sont composées à celle des AF, et c'est l'agent PJ qui définit à quel AF la requête est orientée.

2.3.2. Agent négociateur *AN* :

La négociation d'un service de réservation met en jeu un *AN* créé par *AF* pour compléter la requête et négocier le service dans le cas d'une requête incomplète.

La figure suivante donne le diagramme d'interactions de classes d'agents relatif à notre système de réservation.



Extension possible à d'autre ressource.

Les agents met en jeu (*salle* et *user*) dans notre système ne sont pas destinés exclusivement à une réservation de salles. Moyennant une extension de leurs services et caractéristiques, les agents *salle* peuvent correspondre à d'autres types de ressources du système, et peuvent dans être le support d'une rédaction coopérative de documents,...

3. Fonctionnement du système :

Au lancement du système les *AF* sont créés, puis chacun initialise ses connaissances à savoir la liste des *AR* qu'il coordonne dans son groupe.

➤ Déclaration d'un nouveau membre :

A la création d'un agent, ces caractéristiques sont diffusées au niveau de tous les *AF* (*AR* se déclare auprès de tous les *AF*) et attend les acquittements lui indiquant dans quels groupes il appartient.

▪ Scénario de la déclaration d'un nouveau membre :

1. chaque *AR* se déclare au moment de sa création aux *AF* du système en signalant l'ensemble de ses caractéristiques.
2. chaque *AF* du système fait la comparaison entre ces caractéristiques et celles du groupe.
3. si elles sont compatibles, *AF* le fait appartenir au groupe, sinon il rejette.
4. après l'ajout du nouveau membre au groupe *AF* met à jour ces connaissances en ajoutant l'ensemble des caractéristiques de la ressource à sa base de connaissance.

➤ Résolution d'une requête :

A la formulation d'une requête de réservation par un utilisateur, l'*AR* correspondant la transmet à travers *PJ*, à tous les *AF* des groupes auxquels il appartient.

L'*AF* effectue un appel d'offre, et attend les propositions des différentes ressources salles. Chaque agent *salle* émet, s'il le désire, une offre en fonction de son planning et de ses caractéristiques.

L'AF recevant toutes les offres, évalue la meilleure offre ; pour évaluer l'offre nous avons utilisé une fonction de satisfaction assez simple. Le meilleur serveur (agent *salle* élu) est celui qui correspond à la plus petite différence entre les caractéristiques exigées et celles proposées. Il s'agira lors du choix du serveur *salle*, de tenter d'optimiser les équipements non utilisés.

Notons ici que notre système de réservation ne prend en compte aucune décomposition de requête.

Exemple de requête : réserver une salle au département d'informatique avec 100 table, 400 chaises.

▪ **Scénario de la réservation d'une salle :**

1. l'agent *user* demande de l'AF3 la résolution de la requête.
2. AF3 vérifie la requête.
3. AF3 envoie la requête aux agents *salle* du groupe.
4. A la réception de la requête, chaque agent *salle* élabore une offre en comparant les éléments de la requête avec l'ensemble de ses compétences et envoie sa réponse à l'AF3.
5. AF3 évalue les réponses et établit des contrats avec agent *salle* dont les réponses contiennent des éléments déterminants dans la résolution de la requête.
6. L'agent *salle* envoie un message à l'AF3, pour lui indiquer que la réservation est effectuée.
7. AF3 envoie la réponse à l'agent *user*.

➤ Si l'agent *user* n'a pas spécifié les équipements nécessaires (nombre de tables, nombre de chaises.), un AN est mis en jeu

▪ **Scénario de requête incomplète :**

1. l'agent *user* demande de l'AF3 la résolution de la requête.
2. AF3 vérifie la requête.
3. AF3 crée un agent négociateur AN, ce dernier recopie la liste des caractéristiques des agents *salle* qu'il retrouve auprès du AF3.
4. AN migre vers le site du *user* pour lui demander les équipements nécessaires.



5. *AN* établit la requête complète et l'envoie au *AF3*.
6. *AF3* détruit *AN*.
7. Reprendre l'étape 3 de la résolution de la requête complète.

▪ **Scénario de la déclaration d'un nouveau membre :**

1. Chaque *AR* se déclare au moment de sa création aux *AF* du système en signalant l'ensemble de ses caractéristiques.
2. chaque *AF* du système fait la comparaison entre ces caractéristiques et celles du groupe.
3. si elles sont compatibles *AF* le fait appartenir au groupe, sinon il rejette.
4. après l'ajout du nouveau membre au groupe *AF* met à jour ces accointances.

▪ **Scénario du retrait d'un membre :**

1. *AR* signale au *AF* son intention de se retirer du groupe.
2. *AF* retire *AR* et met à jour ces accointances.

▪ **Scénario de l'évolution d'un agent :**

1. *AR* ayant évolué signale le changement de certains de ses caractéristiques à ses fédérateurs
2. après que chaque *AF* vérifie les caractéristiques, l'*AR* peut être ajouté, retirer ou garder dans le groupe

4. présentation des mécanismes de coopération utilisés :

Le système proposé tente de mettre en œuvre un ensemble de mécanismes parmi tous ceux existant dans la littérature ; nous présentons dans la suite les mécanismes utilisés

4.1. collaboration par appel d'offre :

Dans ce travail, une résolution distribuée d'une requête est le résultat de l'interaction coopératif entre les agents ressources et fédérateurs. Nous avons, en effet utilisé le modèle d'appel d'offre.

Dans ce cas, quatre types de messages sont échangés entre l'agent fédérateur d'un groupe et ses agents ressources lors d'une résolution de requête :

- Message de requête (demande d'offre).
- Messages d'offre de services (élaboration d'offre).
- Message contrat (spécifiant le service élu).
- Message résultat (vers l'agent client)

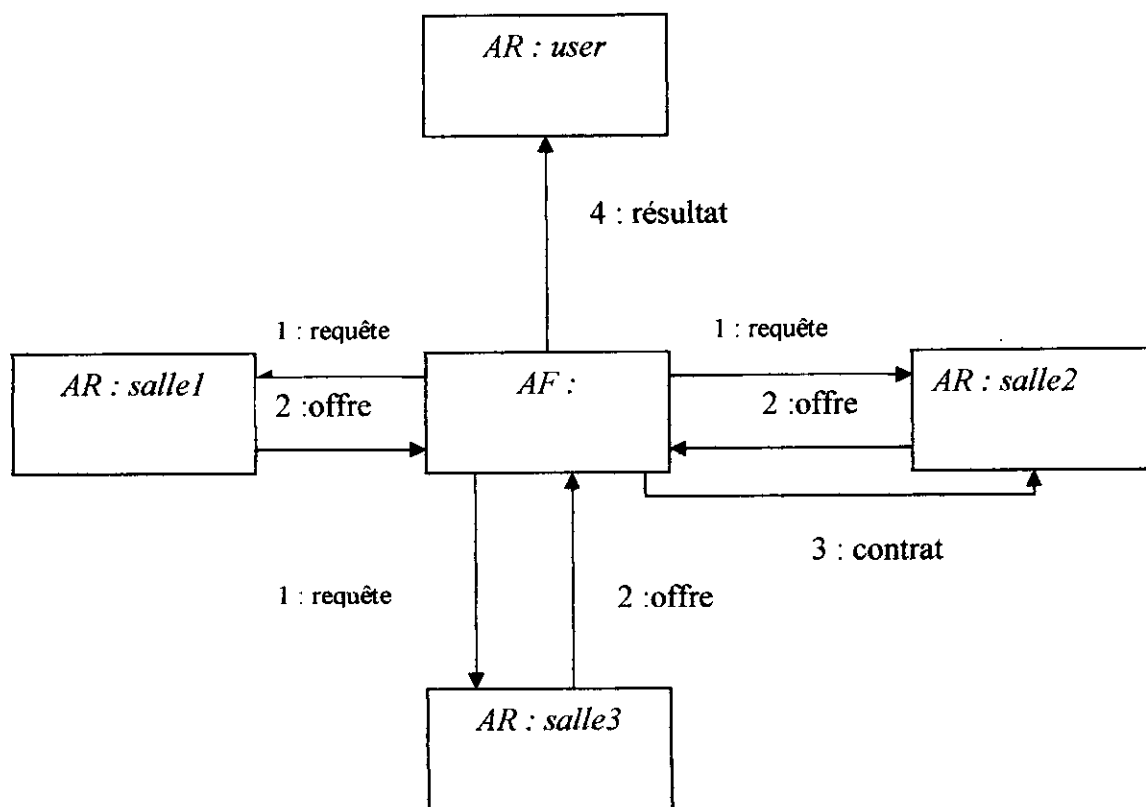


Figure 2.8 : diagramme de collaboration par appel d'offre du système de réservation

4.2. la communication :

Nous avons adopté le mode de communication par envoi de messages, ce mode de communication repose explicitement sur les mécanismes d'envoi/réception de message et peuvent d'avoir une organisation totalement distribuée contrairement à la communication par partage d'information, qui aurait imposé une centralisation des plannings de réservation.

4.3. la coordination d'action :

Dans notre conception, la coordination d'actions se fait par le biais d'agents fédérateurs, qui se chargent de coordonner la réception des requêtes de la part de l'utilisateur et leurs transmissions aux agents serveurs compétents pour la résolution.

Conclusion nous avons présenté dans ce chapitre la conception d'un système multi-agents pour la coopération de ressources dans un SI coopératif.

L'approche de coopération proposée était basée sur les agents, elle bénéficie des avantages offerts par le paradigme agent (réutilisation, modularité, autonomie,...).

Concernant l'étude de cas le langage d'expression de contraintes sur les requêtes est très simple, il permet d'exprimer l'appartenance ou non d'une caractéristique à un intervalle pour calculer la fonction de satisfaction pour élire le meilleur serveur.

On pourrait penser à étudier le fonctionnement sur des situations où les caractéristiques de service peuvent avoir des valeurs différentes (par exemple un agent imprimante, pourrait offrir plusieurs qualités d'impression).

Il faudrait alors engendrer toutes les solutions et évaluer les satisfactions correspondantes.

Dans le chapitre prochain nous donnons des détails techniques sur l'implantation du système.

Chapitre II

Réalisation

Afin d'implémenter les agents de notre système, nous avons opté pour la plateforme *AgentBuilder Pro* version 1.4. vue sa disponibilité sur le marché.

Nous avons utilisé l'environnement ECLIPSE pour la création des interfaces des agents du système. Et pour les bases de données nous avons choisi MYSQL.

I. Description des outils utilisés :

I.1. AgentBuilder :

AgentBuilder est une suite intégrée d'outils permettant de construire des agents intelligents, il a été développé en JAVA, il est composé :

- D'un langage orienté agent, permettant de définir des croyances, des engagements et des actions.
- D'une interface graphique.

Il permet également de définir des ontologies et des protocoles de communications inter-agent, il utilise le KQML comme langage de communication entre les agents.

Un agent créé avec cet outil est typiquement un agent interface, chargé de faciliter la recherche d'information ou la réalisation de certaines tâches à la place de son utilisateur. Un tel agent sera capable de filtrer l'information, de négocier des services avec d'autres agents et de dialoguer avec son utilisateur [AGE1].

AgentBuilder fournit des outils pour :

- Contrôler et organiser le développement du projet.
- Analyser le domaine du problème.
- Spécifier les protocoles d'interaction.
- Définir l'architecture de la coopération.
- Spécifier le comportement de l'agent.
- Créer l'agent exécutable.

Tous les composants de l'AgentBuilder sont mis en oeuvre dans la Java. Cela signifie que le développement d'agent peut être accompli sur n'importe quelle machine ou le système

d'exploitation qui soutient la Java et a un environnement de développement de la Java. De même, les agents créés avec AgentBuilder sont des programmes de la Java donc ils peuvent être exécutés en une JVM (Java virtual machine) [AGE1].

I.2. Eclipse

Eclipse est un projet open source à l'origine développé par IBM pour ces futurs outils de développement et offert à la communauté. Le but est de fournir un outil modulaire capable non seulement de faire du développement en java mais aussi dans d'autres langage et d'autres activités.

I.3. MYSQL :

MYSQL est une base de données, performante, disponible sur de nombreuses plates formes, et en particulier sous Windows et sous Linux.

II. Création d'un agent avec AgentBuilder :

Pour crée un agent avec AgentBuilder il faut :

1. Créer un nouveau projet.
2. Créer une agence ; une agence regroupe les agents qui communique et coopèrent les uns avec les autres pour exécuter quelque tâche. Les agents peuvent être identiques ou spécialisés pour l'exécution de fonctions différentes.
3. Crée l'agent.
4. Crée les règles de l'agent.

II.1. création d'un projet :

Avant la création d'un agent il faut d'abord créer un projet pour gérer le processus de développement d'agent. Cliquer sur **File** puis **new** dans le menu de la première fenêtre de l'AgentBuilder Cela montrera la boîte de Dialogue de Propriétés de Projet. Entrez le nom et la description du projet et Cliquer sur **OK**.

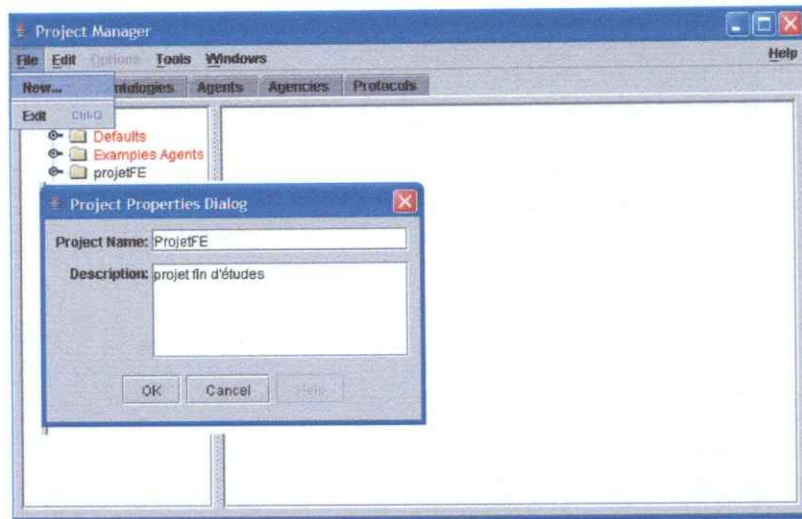


Figure 2.9 : création d'un projet.

II.2. création de l'agence :

Sélectionner l'icône de dossier de projet déjà crée et cliquer avec le bouton droit de la souris. Cela montrera un menu contextuel cliquer sur new **agency** la boîte de Dialogue de Propriétés de l'agence est apparue entrer le nom et la description de l'agence et cliquer sur **OK**.

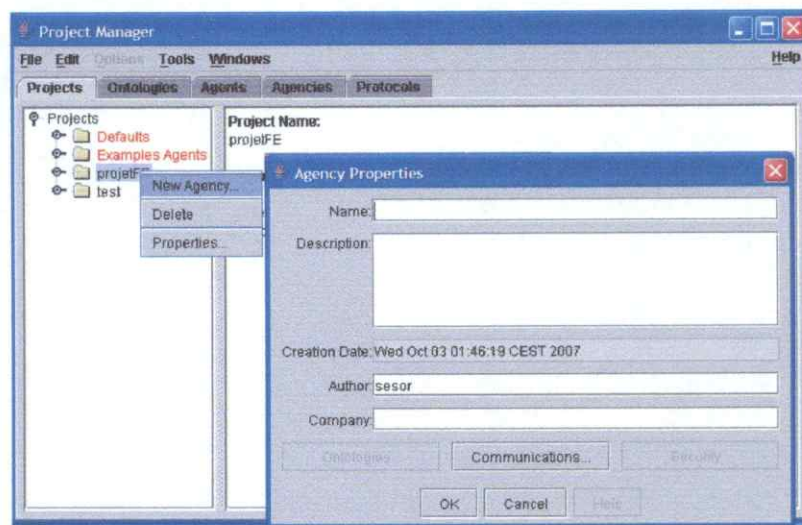


Figure 2.10 : création de l'agence.

II.3. création de l'agent :

De la même façon sélectionner l'icône de dossier de l'agence, cliquer avec le bouton droit de la souris. Cela montrera un menu contextuel cliquer sur new **agent** la boîte de Dialogue de Propriétés de l'agent est apparue entrer le nom et la description de l'agent et cliquer sur **OK**.

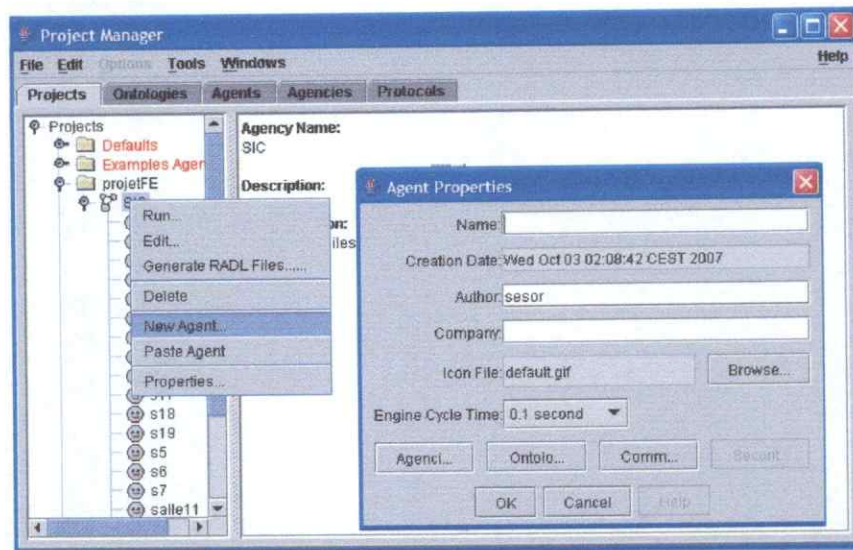


Figure 2.11 : création de l'agent.

II.4. création des règles :

Les règles sont divisées en deux parties ; partie1 condition et partie2 action, pour crée une règle d'un agent il faut le sélectionnée d'abord ensuite cliquer sur l'étiquette **Agents** de la fenêtre Project Manager puis cliquer sur **Tools** ensuite sur **Rule Editor** la fenêtre **Rule Editor** s'ouvre vous pouvez entrer la règle la partie condition dans **Left-Hand Side** et la partie action dans **Right-Hand Side**.

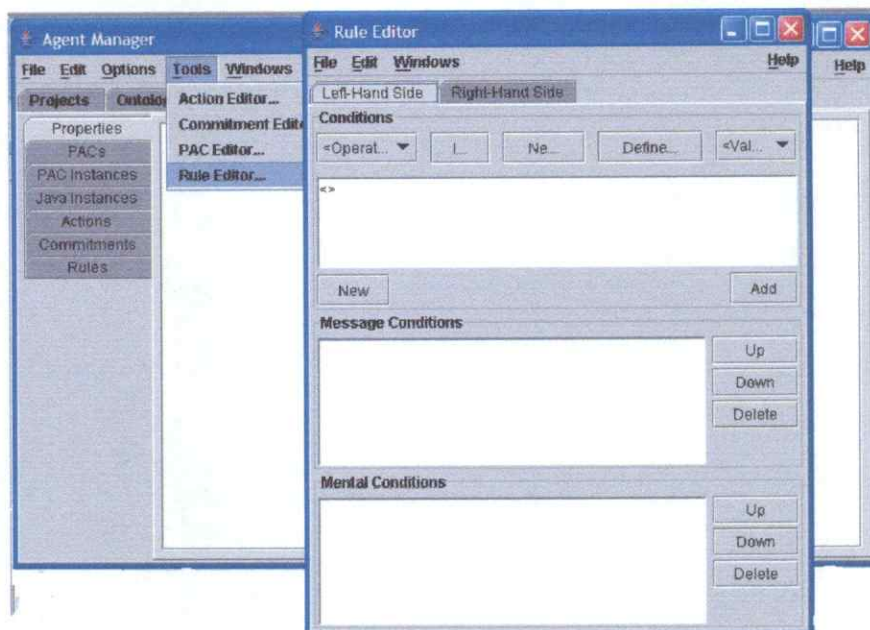


Figure2.12 : création de la règle.

Généralement les actions d'une règle sont des fonctions définis dans des classes java, nous avons utiliser java eclipse pour la construction des classe manipuler par les agents.

III. Réalisation :

Notre application consiste à implémenter le SMA de telle façon à :

- Montrer toute l'interaction et messages échangés entre agents lors du lancement d'une requête.
- Donner la possibilité aux agents de lancer des requêtes (requête locale ou globale).

Ex : faire des consultations des réservations

Pour la réalisation de notre SMA il faut :

- Créer les agents du système.
- Créer les bases de règle et de connaissance des agents.
- Associer à chaque agent une interface permet de lancer des requêtes (requêtes locales ou globales) ou faire des consultations sur la ressource correspondante.

Au lancement d'une requête globale un agent fédérateur doit être sélectionné, nous avons ajouté un nouveau agent appelée «agent page jaune» ; responsable de la sélection d'agent fédérateur apte à satisfaire la requête.

Notre système est composé d'agents fédérateurs, d'agents ressources (salle, user et ressources matérielles) d'un agent intermédiaire de type «page jaune » et un agent interface.

1. L'agent page jaune :

L'agent page jaune doit associer aux différentes requêtes les agents fédérateurs qui sont capables d'y répondre, pour cela les AF doivent s'annoncer auprès de lui, en lui envoyant un message de ses caractéristiques et du type de service qu'ils sont capables de coordonner.

Lorsque un AR client demande à l'agent page jaune s'il y a un agent capable de satisfaire sa requête, celui-ci répond en lui donnant l'AF apte à prendre en charge la requête.

2. L'agent interface :

L'agent interface est ajouté pour donner l'accès aux agents du système via son interface; à la sélection d'un agent, l'agent interface envoie un message à l'agent concerné ce dernier ouvre son interface, pour permettre d'envoyer des requêtes ou de faire des consultations sur la ressource.

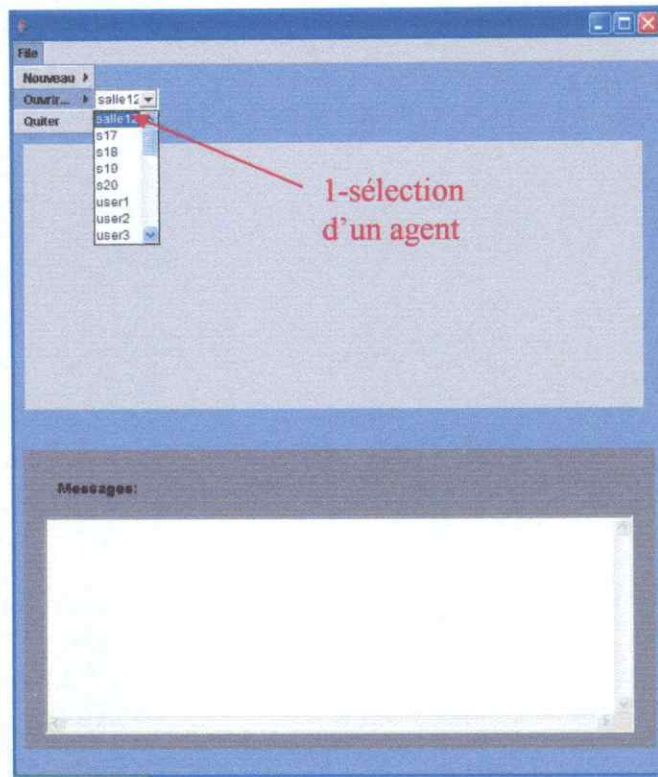


Figure 2.13 : La fenêtre gère par l'agent interface.

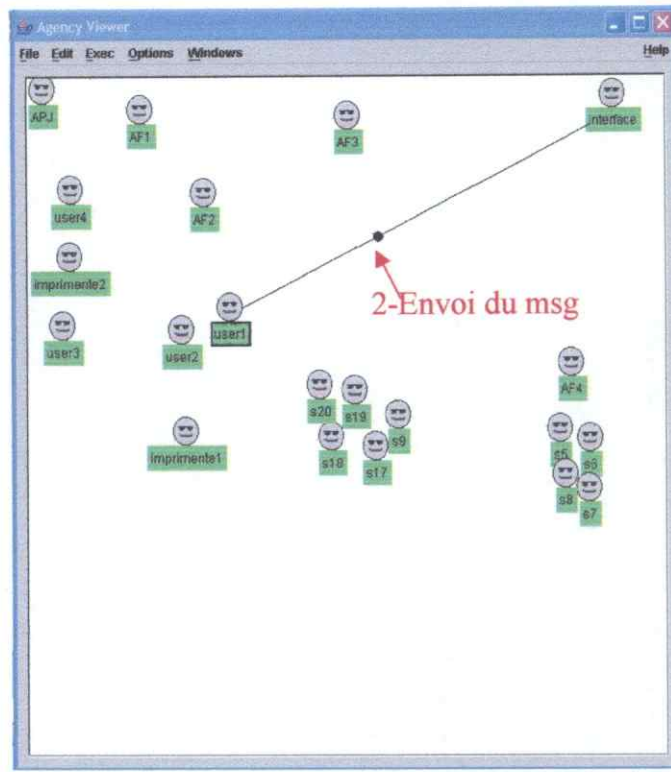


Figure 2.14 : L'envoi du message à l'agent sélectionné.

3. Les fenêtres dans notre réalisation :

Chaque fenêtre de notre réalisation est une interface d'un agent, l'ouverture de cette fenêtre est invoquée par l'envoi d'un message émis par l'agent interface.

Chaque fenêtre est divisée en deux parties :

- **Partie 1** : est spécialisée pour spécifier les caractéristique des requêtes et leurs lancement .
- **Partie 2** : est une zone d'affichage des messages échanger entre les agents de notre système.

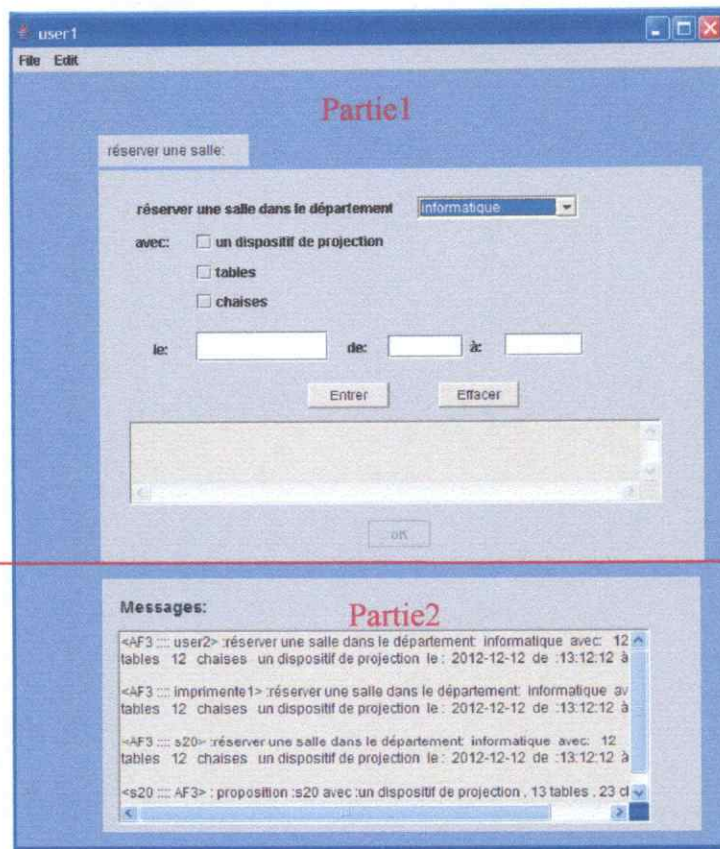


Figure 2.15 : Les partie d'une fenêtre.

Chaque fenêtres de notre système donne la possibilité de consulter les propriétés de l'agent correspondant ; par exemple voir tous les agents appartenant à un groupe (pour un agent fédérateur) ou voir tous les fédérateurs d'un agent ressource, consulter les réservations (pour les agent ressource salle et user), voir les caractéristiques des agents...etc. L'interface de l'agent permet La consultation de ses caractéristiques.

Ces consultations représentent des requêtes locales. Elle permet aussi de lancer les requêtes globales comme la réservation d'une salle ou l'impression d'une liste des réservations....

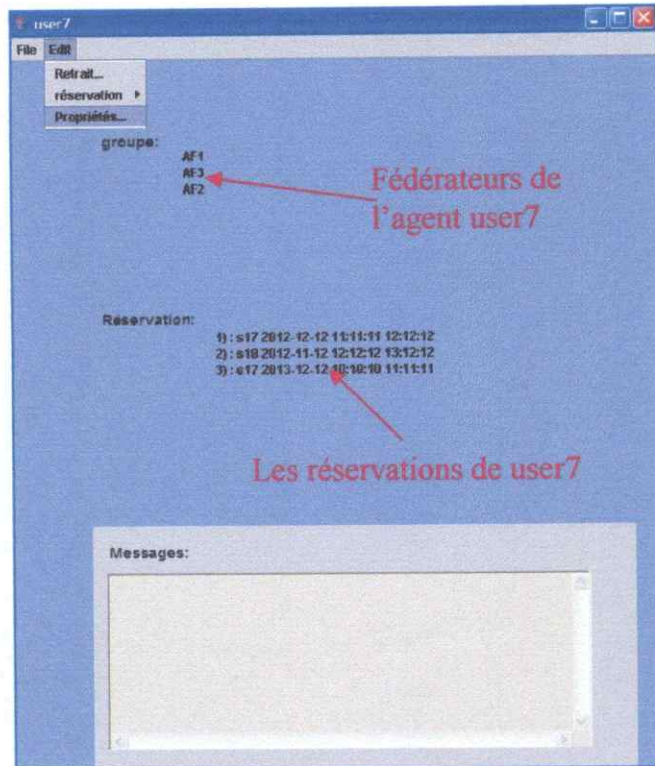


Figure 2.16 : Exemple d'une requête locale consultation de groupe
Et des réservations faite par user7.

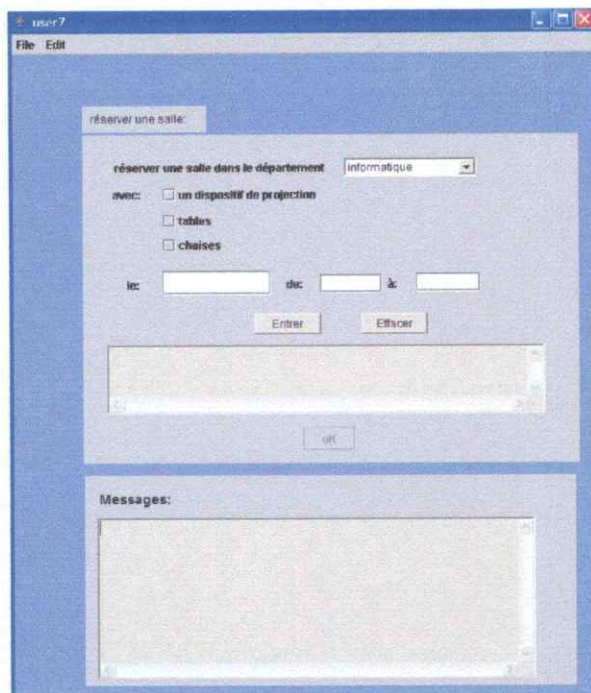


Figure 2.17 : Exemple d'une interface permet de lancer un requête globale
(Réservation d'une salle)

Conclusion générale

Le besoin d'échanger et de partager de l'information est devenue, plus qu'un besoin, une absolue nécessité pour répondre aux contraintes d'une nouvelle économie et aux habitudes engendrées par les nouvelles technologies de l'information.

La diversité de sources d'informations est un frein à la mise en place d'une coopération entre différents systèmes : il s'agit de trouver le moyen d'identifier les sources et les informations pertinentes, puis de les utiliser pour les besoins d'une application spécifique.

L'objectif de ce mémoire est de proposer une solution multi-agent à l'intégration de ressources dans un système d'information distribué coopératif afin d'assurer un accès transparent aux informations et aux services de cette coopération.

L'approche proposée cherche à répondre à des besoins importants dans la conception de SI coopératif comme la flexibilité et la facilité de l'évaluation dans un environnement distribué où les ressources sont en perpétuelle évaluation.

Notons néanmoins quelques limites qui peuvent ouvrir des voies vers de nouvelles perspectives :

- Le langage d'^{expression} exception de contraintes sur les requêtes est très simple. Il permet d'exprimer l'appartenance ou non d'une caractéristique à un intervalle pour calculer la fonction de satisfaction pour élire le meilleur serveur. Il faudrait songer à un langage plus élaboré qui peut prendre en compte toutes les solutions possibles.
- Il faudrait penser à la décomposition de requêtes n'a pas été considérée dans le système actuel, la collaboration à la résolution d'un but global impose impérativement une telle décomposition. Il faudrait donc en perspective penser aux mécanismes adéquats de décomposition de requêtes et de recombinaison de résultats correspondants
- Prendre en charge le protocole de coopération de manière à pouvoir rendre le système opérationnel. Ceci, revient à développer encore plus le mécanisme de négociation qui est à la base de toute coopération (voir du côté de la FIPA/ACL).

Nous ne pensons pas avoir donné un système complet opérationnel, nous pouvons avoir surtout réussi à approcher beaucoup de problèmes qui sont au carrefour de plusieurs domaines :

- Les systèmes d'information
- Les systèmes multi-agents.
- Les systèmes distribués

Nous avons cherché à montrer les potentialités des SMA en matière de coopération dans un environnement ouvert de ressources hétérogènes et évolutives.

La configuration des différents groupes d'agents se fait dynamiquement et de façon intelligente. A un niveau conceptuel l'agent représente dans notre approche toute entité potentiellement cliente et/ou prestataire de services disponibles dans le SI coopératif (ressources matérielles et logicielles, intervenants humains,...etc).

Nous trouvons les agents particulièrement intéressants comme outils d'abstraction en regard avec la complexité d'intégration d'un nombre important de ressources dans une organisation commune. Nous avons présenté une modélisation de la couche de coopération à l'aide d'un modèle agent et défini un processus de résolution d'une requête globale basé sur ce modèle.

L'outil développé est loin d'être un logiciel fonctionnel pour le marché de la coopération, c'est plutôt un prototype pédagogique qui permet de mettre en œuvre des aspects de réflexion sur la coopération de ressources dans un système d'information distribué.

BIBLIOGRAPHIE

[ADA 00] Emmanuel ADAM « Modèle d'organisation multi-agent pour l'aide au travail coopératif dans les processus d'entreprise : application aux systèmes administratifs complexes » UNIVERSITÉ DE VALENCIENNES ET DU HAINAUT-CAMBRÉSIS 2000

[AGI 96] Agimont, G. (1996). *Modélisation et Simulation des Organisations Multi-agents*, Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis.

[AUS 62] John.L Austin, « How To Do Things with Words », Clarendon Press, 1962

Bannon L. (1997) : Group Decision Support Systems : An analysis and critique. Actes de la Conférence Internationale en "Information Systems", Cork, volume 1, pp 526-539.

[BEN 99] BENSLIMENE.D « systemes d'information cooperatifs : une approche à base de mediation et d'agents » univesite Bourgogne 1999

[BRA 87] M. E. Bratman. « *Intention, Plans, and Practical Reason* ». Harvard University Press, Cambridge, MA, 1987.

[BUK 97] O.BUKHERS. A.Elmagarmid " object oriented multidata base systems: asolution for advanced applications". Prentice Hall.1996.

[BRA 88] M. E. Bratman, D. Israel, M. E. Pollack. « Plans and resourcebounded practical reasoning », Computational Intelligence 4 : 349-355. 1988.

[BRO 86] R. A. Brooks. « *A robust layered control system for a mobile robot* ». *IEEE Journal of Robotics and Automation*, RA-2(1) :14-23, 1986.

[BRO 91] R. A. Brooks. « *Intelligence without representation* ». Artificial Intelligence , 47(1-3) :139-159, 1991.

[BON 94] E. Bonabeau, G. Theraulaz, «*Intelligence collective*».Hermès, Paris, 1994.

[BON 99] E. Bonabeau, M. Dorigo, G. Theraulaz, «*Swarm Intelligenc : From Natural to artificial Systems* ». Oxford University Press, 1999.

[BAE 96] Baeijs, C. and Demazeau, Y. (1996). Les organisations dans les systèmes multi-agents. In *5ème Journée Nationale du PRC-IA sur les Systèmes Multi-Agents*, pages 35–46.

[BOU 93] : Structure de communication et d'organisation pour la coopération dans un univers multi-agents. Rapport du Laforia n°93/04, Février.1993.

[BAL 06] Bala mahfoud « Interopérabilité Sémantique des Systèmes d'Information Distribués » institut national d'informatique (INI) INI, Oued Es-smar, Alger

[BOO 91] BOOCH G, object oriented design with applications, the Benjamin/ Cumming Publishing Company, Inc 91

[BOU 00] MOKRANE BOUZEGHOUB, GEORGE GARDARIN, PATRICK VALDURIEZ, "les objets" Edition revues et augmentée EYROLLES 2000

[BOU 56] BOULDING.K « general systems theory the skeleton of science » , Management Science 1956.

[BRI 01] BRIOT.J, DEMAZEAU.Y, « principes et architecture des systemes multi-agents » Lavoisier 2001.

[BRA 96] BRADSHAWJ.M, DUTFIELD.S, BENOIT.P, WOOLEY.J.D " KAOS: toward an industrial-strength generic agent architecture" AAAI/MITPRESS 1996.

[BRA 87] BRATMAN.M.E, "Intention plan and practical reason", Harvard University Press 1987.

[BUS 99] BUSSE.B « federated information systems : concept, technology and architectures » Berlin university 1999.

[CHA 01] Chaib-Draa B., Jarras I., Moulin B., "*Systèmes multi-agents : principes généraux et applications*" Principes et architectures des systèmes multi agents, J.-P. Briot et Y. Demazeau (Eds.) Hermès.

[CHA 94a] CHAIB6DRAA.B, LEVESQUE.P, « hierarchical models and communication in multi-agents environnements » Marcel Dekker 1994

[CAS 98] CASTELFRANCHI.C, CONTE.R, "distributed artificial intelligence and social science: critical issues" Angeleterre 1996

[COH 90] COHEN.P.R, LEVESQUE.H.J, "intention is choice with commitment» IA 1991.

[CAM 90] CAMPBELL.J, DINVERNO.M "knowledge interchange protocol" Elsevier science publishers 1990.

[D.MI 94] De Michelis G., From the analysis of cooperation within work-processes to the design of CSCW Systems. *Proceedings of the 15th Interdisciplinary Workshop on Informatics and Psychology: Interdisciplinary approaches to system analysis and design*, (24-26 May 1994, Schaerding, Austria).

[DEL 93] DELATTE B, HEITZ M, MULLER J.F, HOOD Reference Manual 3.1, Edition Masson 93.

[D.HE 01] D.HERIN et al, « des systemes d'informations cooperatifs aux agents informationnels » Ingenierie des Systemes d'Informations. HERMES 2001

[DRO 98] DROGOULA, FRESNEAU.D, "métaphore du fourragement et modele d'exploitation collective de l'espace sans communication ni interaction pour les colonies de robots autonomes mobiles » 1998.

[DEM 91] DEMAZEAU.Y, MULLER.J6P,"from reactive to intentional agents" dans Decentralized AI, Elsevier Science Publishers.B.V 1991

[DAV 83] DAVIS.R, SMITH.R, "negotiation as a metaphor for distributed problem solving" AI 1983.

[DUR 87] DURFEE.E.D, LESSER.V.R, CORKILL.D, "cooperation through communication in a distributed problem solving network" DAI Pitman 1987.

[ESP 95] ESPINASSE B..SPINOZA L.M..CHOURAQUI E.."D-CIM et IAD: une approche orientée connaissance pour la modelisation des systemes de production" congrés international de genie industriel de montréal: la productivité dans un monde sans frontière. Montréal. Canada. 1995

[FER 92] I. A. Ferguson. « *TouringMachines : An Architecture for Dynamic, Rational, Mobile Agents* ». PhD thesis, Clare Hall, University of Cambridge, UK, November 1992.

[FIK 71] R.E. Fikes et N.J. Nilsson, « *STRIPS: a new approach to the, application of theorem proving to problem solving* ». Artificial Intelligence, 2 (3-4):189--208. 1971.

[FIS 99] K. Fischer, B. Chaib-draa, H. J. Müller, J. P. Müller, M.Pischel.
« *A simulation approach based on negotiation and cooperation between agents* ». *IEEE Trans. on Systems, Man, and Cybernetics*, 29(4), pp. 531-545, 1999.

[FAK] HOUSSAM FAKIH « les protocoles d'interaction dans des SMA »

[Fox, 1981] Fox, M. (1981). An organizational view of distributed systems. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 11, pages 70–80.

[FER 95] FERBER.J, “ les systèmes multi agents vers une intelligence collective”, InterEditions 1995.

[FRA 97] FRAMKLIN.S, GRAESSER.A “ is it an agent, or just a program? A taxonomy for autonomous agents” Heidelberg Allemagne 1997.

[GLI 97]: P. GLIZE « Coopérer pour apprendre sans présupposer. *Dans Conference on Learning : from Natural Principles to Artificial Method*” Genève, Suisse, 1997.

[GEN 98] GENESERETH.M.R, “knowledge interchange format” Stanford university, California 1998.

[GEL 92] GELERNTER.D, CARRIERO.N, “coordination languages” communication of the ACM 1992.

[GUA 98] GUARINO.N, « formal ontology and information systems ». conference on formal ontology in information system . Torino 1998

[HER 01] HERIN.D « des systèmes d’informations coopératifs aux d’agents informationnel » ISI, HERMES 2001.

[HEW 77] HEWITT.C, « viewing control structures as patterns of passing messages” , artificial intelligence 1977.

[HUG 05] Marc-Philippe Huet, « *Agent Communication* », In Intelligent Decision support systems in Agent-Mediated Environments, vol 115, frontiers in Artificial Intelligence and Applications. Gloria Phillips-wren and Lakhmi Jain (eds), IOS Press, 2005.

[HUG 02] Huet M.P, “ Agent UML class diagrams revisited” in proceeding of agent technology and software engineering (AgeS)

[HUB 02] Hübner, J. F., Sichman, J. S., and Boissier, O. (2002). A model for the structural, functional, and deontic specification of organizations in multiagent systems. In *16th Brazilian Symposium on Artificial Intelligence*, pages 118–128.

[JAC 92] JACOBSON.I, CHRISTERSON.M, JONSON.P, OVERGARD.G, object oriented soft engin , ause case driven approach, Addison-wesley 92.

[JEN 98] JENNINGS.N.R, WOOLDRIDGS.M “ a roadmap of agent research and development “ InterEditions 1998.

[KRI 91] KRICHNAMURTHY.R, « langage feature for interoperability of database with schematic discriponcies » ACM sigmod 1991

[KHOS 98] KHOSHAFIAN.S , BUCKIEWIEZ.M, « groupware & workflow » ,InterEditions 1998

[LEM 74] J.L Lemoigne. « les systèmes d'informations dans les organisations » PUF 1974

[LIT 86] Litwin. W et ADBELATIF A « multidatabase interoperability » IEEE computer, vol 19,n 12 1986

[LIU 95] LIU.L , LU.C , « the distributed interoperable object model an dits application to large-scale interopepable database systems » international conference on information and knowledge management.1995

[LIU 02] Y.J. Liu, « *De la nécessité et de la façon de coopérer, de s'autoorganiser et de se reconfigurer dans des systèmes de production complexes : modélisation et gestion d'un système virtuel pour une approche multi-agents* », Thèse de Doctorat en Génie Industriel, INPG, France, 2002.

[MAL 87] Malone, T. (1987). Modeling coordination in organizations and markets. In *Management science*, volume 33, pages 1317–1332.

[MAR 95] Marshak R. T., Workflow : Applying Automation to Group Processes. *Groupware, Technology and Applications*, Ed. by Coleman D. & Khanna R., Prentice Hall, pp. 71-95, 1995.

[MAR 58] March, J. and Simon, H. (1958). *Organizations*. New-York.

[M.FA 02] M.FAKIH et Z.GUESSOUM “les protocols d'interaction dans des systemes multi agents” papport de DEA. LIP6 2002

[MIC 97] MICHELIS.G, « cooperative information system : a manifesto, cooperative information system” academy PRESS 1997.

[MIN 82] Mintzberg, H. (1982). *Structure et Dynamique des Organisations*. Organisation.

[MUL 94] J.P. Müller et M. Pischel. « *Modelling interacting agents in dynamic environments* ». In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94)*, Amsterdam, The Netherlands, pp. 709-713, 1994.

[MOU 96] MOULIN.B, CHAIB6DRAA.B, “an overview of distributed artificial intelligence” 1996.

[NUM 90] NUMAOKAC, TOKORO.M, “conversation among situated agents” dans M.N.HUHNS(dir) , tenth international workshop on distributed AI 1990.

[OUA 04] OUAHRANIL, « agents pour la coopération de systèmes d’information » MCSEAI.Tunisie 2004.

[ODE 00] J. ODELL, H. VAN DYKE PARUNAK et B. BAUER : Extending UML for Agents. *Dans Proceedings of the Agent-Oriented Information Systems (AOIS) Workshop at the 17th National Conference on Artificial Intelligence (AAAI)*, pages 3–17, 2000.

[ODE 01]J. ODELL, H. VAN DYKE PARUNAK et B. BAUER : Representing Agent Interaction Protocols in UML. *Dans First International Workshop, AOSE 2000 on Agent-Oriented Software Engineering*, pages 121–140. Springer-Verlag, 2001.

[PAP 92] PAPAZOGLOU.P.P « an organisationnal framework for operating intelligent information systems » international journal of intelligent and cooperative information system 1992

[PAS 05] Pascale Zaraté «Des Systèmes Interactifs d’Aide à la Décision Aux Systèmes Coopératifs d’Aide à la Décision : Contributions conceptuelles et fonctionnelles » INSTITUT DE RECHERCHE EN INFORMATIQUE DE TOULOUSE (IRIT) 2005.

[RAO 91] RAO.A.S, GEORGEFF.M.P, “modelling rational agent within a BDI-architecture” 1991.

[ROS 86] ROSENSCHEIN.S.J, KAEHLING, “ a situated view of representation and control “ Morgan Kaufmann 1986.

[RUM 91] RUMBAUGH.J, BLAHA. M, PREMERLANJ.W, EDDY.F, object oriented modelling and design, Prentice Hall International 91.

[SCH 97] Schael T., Théorie et pratique du workflow. Des processus métiers renouvelés. New-York : Springer Edition, 1997

[SCI 94] SCIORE.E “ using semantic values to facilitate interoperability among heterogeneous information systems”, ACM Transactions on Database Systems, 1994.

[SEA 69] J. Searle, “*Speech Acts: An Essay in the Philosophy of Language*”; Cambridge University Press, Cambridge, 1969

[SHE 90] SHETH A., LARSON J. Federated database systems for managing distributed, heterogeneous, and autonomous databases, ACM Computer Surveys 22, 3 Sept. 1990

[SHE 92] SHETH.A, so far (schematically) yet so near (schematically), in proceedings of the IFIP ds-5, conferences on semantic of database system; 1992

[SHO 93] SHOHAM.Y, agent oriented programming », artificial intelligence, vol 60, n 1 Elsevier 1993.

[SHE 99] Semantic Information Brokering: How Can a Multi-Agent Approach Help? Amit Sheth, Vipul Kashyap, and Tarcisio Lima University of Georgia, Athens GA

[TAR 83] TARDIEU.H, ROCHFELD.A, “la méthode MERISE: principes et outils” les éditions d’organisation, PARIS 1983.

[THO 95] THOMAS.S.R, “ the placa agent programming language”, springer-verlag 1995.

[TOU 06] TOUNSI Jihène « MODÉLISATION ET SIMULATION SMA DU SYSTÈME DE PILOTAGE APPLIQUÉ AUX SYSTÈMES DE PRODUCTION » Rapport de Master 2 en Sciences et Technologies- université SAVOIE 2006

[WIL 94] Wilson P., Introducing CSCW, What it is and why we need it. *CSCW : The multimedia and networking paradigm*, UNICOM, ed. by S.A.R. Scrivener, Aldershot, England: Avebury Technical, pp. 1-18, 1994.

[WOO 02] Mike Wooldridge, “*An Introduction to MultiAgent Systems*”, John Wiley & Sons, Février 2002.

[WOO 98] A Roadmap of Agent Research and Development - Jennings, Sycara, Wooldridge (1998)

Webliographie

[AGE1] – www.agentbuilder.com

[AGE2] <http://www.agentware.com/main/tec/whitepaper.html>

OMG. The Object Management Group, <http://www.omg.org/>

AUML. The Agent Unified Modelling language, <http://www.auml.org/>

[FIPA, 1997a] Foundation for Intelligent Physical Agent. Part 1: Agent Management <http://www.fipa.org> Version 1.2, 1997

Liste plates-formes multi-agents <http://www.agentbuilder.com/AgentTools/>

KQML <http://www.cs.umbc.edu/kqml/>