Quelques perspectives d'utilisation des Langage et Protocoles d'Interaction dans le contexte des Télécommunications

Jean-Luc Koning (LIFIA-ISAR), Yves Demazeau (LIFIA-CNRS)
koning@isar.fr, Yves.Demazeau@imag.fr
46 avenue Félix Viallet F-38031 Grenoble cx

Babak Esfandiari (LIRMM), Joel Quinqueton (INRIA-LIRMM)
esfandiari@lirmm.lirmm.fr, jq@lirmm.lirmm.fr
161 rue Ada F-34392 Montpellier Cedex 5

Abstract

Au sein de la communauté IAD, il est habituellement acquis que la coopération entre des agents intelligents requiert une communication explicite, et que le terme de 'communication' signifie davantage que ce qu'il représente dans les systèmes distribués traditionnels. Dans cet article, nous proposons un "langage d'interaction", inspiré de la théorie des actes du langage et préconisons son utilisation dans le cadre de la supervision de réseaux de télécommunications. Un tel langage peut servir non seulement à faire interagir différents agents et superviseur d'un réseau de télécommunications, mais également à modéliser les différentes interactions entre rôles primitifs qu'un agent du réseau peut assurer. Aller ainsi d'une interaction entre agents vers des interactions entre primitives d'un agent, en utilisant un meme formalisme et un même outil, nous semble constituer un premier pas essentiel vers des langages orientés agent qui soient de réels langages de programmation.

1 Introduction

La plupart des approches d'IAD pour communiquer entre agents sont fondées sur la théorie des actes du langage introduite par Austin et Searle [16]. Dans un acte du langage, on fait la distinction entre les aspects *locutoires*, *illocutoires*, et *perlocutoires*. La profération d'une phrase (acte locutoire) se distingue de son effet attendu (acte illocutoire) et de l'effet réel (acte perlocutoire) sur l'auditeur. La classification des actes illocutoires en assertifs, directifs, déclaratifs et expressifs par Searle a conduit les chercheurs en IAD à choisir la nature des types de messages. *Il nous semble que les actes illocutoires peuvent explicitement être encodés dans le message en termes de type de messages.*

Egalement, en réponse aux difficultés posées par la classification de tous les actes du langage en Allemand, Ballmer et Brennenstuhl [1] ont suggéré de regrouper les verbes dans des catégories sémantiques, et ont regroupé ces catégories sémantiques en modèles, en incluant un ordre de ces catégories selon leur relations temporelles et selon leur degré de force. Ces catégories et modèles ont conduit à des dialogues et des cadres appropriés pour structurer des *protocoles* de dialogue. Une telle idée a guidé un grand nombre de chercheurs en IAD à concevoir des protocoles ad hoc tels que pour la négociation par exemple. *Nous allons développer davantage cette idée en introduisant des "Protocoles d'Interaction" génériques qui peuvent ensuite être instanciés pour implanter différents comportements sociaux ad hoc.* De tels Protocoles d'Interaction sont habituellement appelés des "Protocoles de Communication" en théorie des actes du langage. Une raison pour laquelle nous préférons utiliser le terme de “Protocole d'Interaction” est pour distinguer clairement ces protocoles des protocoles de communication standard qui existent dans les systèmes distribués.

2 Vers un cadre unifié pour l'échange de connaissances

2.1 Le Langage d'Interaction

Comme cela est indiqué dans [3], si la communication est simplement décrite en termes d'émission et de réception de messages, chaque agent doit être capable d'inférer l'intention de l'émetteur lorsqu'il énonce le message. Si les messages ne sont pas structurés, cette inférence peut être très difficile, voire impossible. C'est pourquoi les messages doivent être attachés à des restrictions formelles, et structurés de façon à en faciliter l'interprétation.

Cela peut être effectué par exemple en employant des types de messages, de sorte que l'intention de l'émetteur soit immédiatement reconnue à partir du message lui-même. Cela nous amène naturellement à faire la distinction entre le *Langage de Communication* et le *Langage de Représentation des Connaissances*. Le premier exprime principalement le message d'un point de vue strictement Systèmes Distribués, alors que le second inclut à la fois la connaissance du domaine multi-agents (*Langage Multi-Agents*, qui véhicule, tant que faire se peut, l'intention de l'émetteur), mais aussi le *Langage du Domaine d'Application* (par exemple, ici, le langage du domaine des Télécommunications).

Le *Langage d'Interaction* développé au LIFIA, introduit dans [8] tel que nous l'utilisons ici, et en cours d'extension dans [9], définit le vocabulaire commun et sa sémantique. Il est utilisé pour construire des échanges d'informations, appelées *interactions*, entre les agents. Les interactions peuvent être des échanges d'actions, de plans, de buts, ou d'hypothèses correspondant aux types des informations que les agents manipulent. Une interaction est construite selon la syntaxe suivante:

<interaction> ::= <communication> <multi-agents> <application>

2.2 Le Langage de Communication

Le langage de communication, décrit par <communication>, consiste en différents champs utilisés par la couche de communication du système.

<communication> ::= <from> <to> <id> <via> <mode>

Ce sont l'émetteur <from> et le récepteur <to> du message (le contenu de ce champ pouvant être soit un identificateur d'agent, soit *broadcast*), l'identificateur du message <id>, le type de médium de communication utilisé <via> (dans certains systèmes, plusieurs types de médium peuvent être utilisés, par exemple l'un à très haut débit pour les importants volumes de données, et un autre pour les simples échanges de messages par mécanisme de boîte aux lettres), et le <mode> de communication (synchrone ou asynchrone).

Cette couche du système est très largement classique dans les langages de télécommunications. Elle permet de spécifier le canal de communication.

2.3 Le Langage Multi-Agents

Il est évident, pour un niveau de compréhension donné, que le degré de complexité d'un protocole d'interaction, et donc du langage multi-agents, est inversement dépendant du degré de développement du langage du domaine d'application qui est utilisé par les agents pour interagir dans la société. Afin d'utiliser un langage du domaine d'application simple, du premier ordre par exemple, pour un domaine comme celui des Télécommunications, nous nous focalisons ici sur un langage multi-agents assez complexe, qui sera associé à des protocoles d'interaction qui peuvent être des plus simples aux plus complexes.

Notre langage multi-agents, décrit par le champ `<multi-agents>`, regroupe toute l'information relative au domaine multi-agents. Les trois types d'information sont le type, la nature, et la force de l'interaction.

`<multi-agents> ::= <type> <nature> <force>`

Le `<type>` de l'interaction est celui proposé dans le domaine des systèmes multi-agents par Gaspar [11]. On le retrouve également dans le domaine des télécommunication à travers le langage CMIS.

La `<nature>` définit à quelle couche de contrôle de l'agent récepteur s'adresse le contenu de l'interaction, et prévoit trois possibilités : couche de décision (cf. buts), couche de commande (cf. actions), couche d'observations (cf. faits ou hypothèses). Pour nous, et tant que ça s'avère possible, un tel champ `<nature>` doit être défini indépendamment du modèle d'agent. Cependant, il est évident qu'il est néanmoins nécessaire de connaître la structure de tous les agents pour remplir adéquatement ce champ.

La `<force>` définit la priorité de cette information, tout en dénotant partie de l'intention de l'émetteur. Les possibilités sont empruntées de la théorie des actes de langage, allant du *commandement*, de priorité maximale, à la simple *information*, de priorité minimale. Cette échelle peut être adaptée, et nous avons adopté comme base l'ensemble des étiquettes illocutoires défini par Campbell et d'Inverno [5], duquel nous avons extrait un sous-ensemble de tons adapté aux types d'interactions auxquelles nous nous intéressons. Ils permettent d'exprimer les intentions suivantes : recherche d'information, information, avertissement, conseil, marchandage, persuasion, commandement, et expression.

Notons, et cela est particulièrement important, que nous pouvons garantir la réception et le traitement de tout message émis en restreignant les possibilités du langage de communication. De fait, il suffit de mettre la `<force>` du message à sa priorité maximale pour que, eu égard aux positions relatives des agents dans la société, l'émetteur dominant le récepteur, il soit admis que l'interaction soit interprétée par le récepteur comme un ordre à satisfaire.

Il va sans dire que pour qu'un message soit interprété comme un ordre il est nécessaire que d'un point de vue social l'agent émetteur ait le pouvoir de donner cet ordre. En d'autres termes, la `<force>` associée à un message ne doit pas être antinomique avec le rôle de l'agent qui le construit. A titre d'exemple, dans le domaine de la vision, nous avons [8] modélisé deux systèmes de vision à l'aide de systèmes multi-agents [8] et avons démontré la convergence du processus en traduisant les ordres du système monolithique sous forme de relation maître-esclave. Ce mode de fonctionnement extrême des protocoles dans ce cas n'a été mis en place que pour montrer la faisabilité de l'approche en se ramenant aux résultats d'un système distribué.

2.4 Introduction aux Protocoles d'Interaction

L'utilisation d'un langage multi-agents permet à chaque agent d'extraire explicitement de la sémantique du message de l'information qui serve à contrôler l'échange d'information et par là même au contrôle de la société d'agents dans son ensemble. Le fait de découpler l'intention de l'expéditeur du message lui-même est un premier pas dans ce sens mais n'est pas suffisant. En effet, les agents doivent aussi savoir comment réagir à un message, ou bien ce qu'ils doivent attendre après avoir envoyé un message. C'est cet ensemble de spécifications auquel il s'agit de répondre dans un cadre unique, permettant ainsi de structurer les interactions entre agents, que nous appelons le protocole d'interactions.

Chaque type de message d'un langage multi-agents est en fait associé à un protocole de base donné, qui dans l'état actuel [8] est choisi par le récepteur du message en fonction du type du message, comme dans le travail de Burmeister [4]. Ce que nous entendons ici par "protocole d'interaction" est ce qu'on définit pour restreindre les interactions entre agents en fonction des problèmes à résoudre.

Il existe un certain nombre de protocoles classiques dans la littérature d'Intelligence Artificielle Distribuée. Citons par exemple celui des *Réseaux Contractuels* [18], ou bien le protocole dit *hiérarchique* [10]. Plus récemment, sont apparus des protocoles plus spécifiques, pour l'introduction de

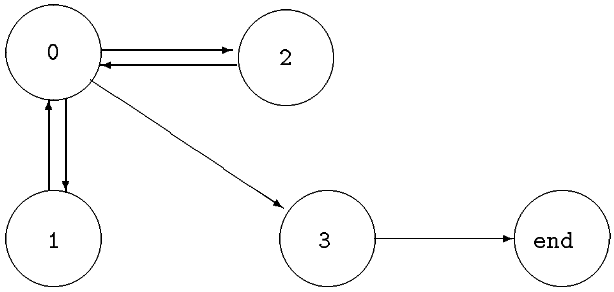

Figure 1: Représentation Générale d'un Protocole d'Interaction

nouveaux agents [7] ou encore la négociation [6]. Ce dernier, basé sur des modèles provenant de l'étude des situations de lutte pour la survie dans les sociétés humaines. Ce protocole prend en considération les situations de conflits où le contrôle est distribué et où aucune solution globale exacte n'existe.

Ce tour d'horizon des protocoles d'échange de connaissances nous permet de jeter les bases d'une maquette générique, qui s'efforcera de les expérimenter dans le domaine de la supervision de Réseau de Télécommunication. Avant de le faire, nous montrons comment utiliser les protocoles de communication, et nous illustrons cette utilisation dans le domaine de la Vision par Ordinateur.

2.5 Contrôler le Déroulement d'un Echange d'Information

Pour [8], un protocole d'interaction est un graphe d'état prédéterminé dans lequel l'agent évolue en fonction des types de messages reçus. Ceci conditionne son comportement, en particulier les types de messages qu'il doit à son tour émettre dans une situation donnée. On peut, grâce à ce type de graphes, modéliser des protocoles connus tels que celui des Réseaux Contractuels [18].

Comme montré sur la figure 1, un protocole d'interaction peut être vu comme un réseau de transitions montrant un ensemble d'états possibles (dans les cercles) dans lesquels les agents peuvent se trouver alors qu'ils inter-agissent (parcourent un arc). Un agent passe d'un état à l'autre selon le type de messages qu'il reçoit. Ceci affecte son comportement et particulièrement le type de messages qu'il doit envoyer dans une certaine situation. Ceci contraint également la valeur de certains champs dans le message. De plus, chaque transition peut être étiquetée d'une condition qu'un agent doit satisfaire avant de la traverser.

La description d'un protocole peut être vue comme un ensemble de règles de production. La condition attachée à une transition doit être satisfaite pour que l'échange d'information se fasse entre les deux agents en jeu. Donc une transition peut s'énoncer de la manière suivante :

`si` état $s1$ `&` condition c `alors` état $s2$

signifie qu'étant donnée une certaine conversation en cours entre deux agents A et B, si l'agent A est dans l'état $s1$ et que la condition c est vraie alors l'agent B passe dans l'état $s2$ de cette conversation quand il reçoit le message de A. La différence fondamentale avec les Réseaux de Petri est que l'état suivant ne peut pas être déterminé par une décision externe, c'est-à-dire, elle n'est pas inférée de l'état courant de l'agent et de l'état général du réseau, mais l'agent lui-même évalue et fait son propre choix, la décision est prise selon une perspective locale.

Bien sûr cela demande que d'une manière ou d'une autre le message contienne explicitement l'état de la conversation auquel les agents sont arrivés lors de la réception du message. Ceci est nécessaire pour éviter la confusion entre plusieurs conversations ayant lieu au même moment. En effet, on peut facilement imaginer un moment dans le processus de conversation entre les agents où B est au milieu de deux conversations différentes avec le même agent A.

2.6 Un exemple d'Utilisation de Protocoles d'Interaction en Vision par Ordinateur

L'intégration de plusieurs modules de vision consiste à les faire travailler ensemble en les considérant comme une société d'agents d'un système de vision. Les interactions entre ces agents de base sont habituellement très primaires dans les systèmes de vision existants. Ceci est principalement dû au fait que la société est composée d'agents hétérogènes qui ont leurs tâches spécifiques bien définies et que la société est souvent organisée hiérarchiquement. Les interactions entre les agents sont limitées à des requêtes sur la représentation de la scène contenue dans chaque agent. Uniquement un ensemble de requêtes de base sont définies pour permettre aux agents de se demander les uns aux autres des informations sur certains types de caractéristiques. Par conséquent, l'architecture du contrôle est une question cruciale pour la flexibilité de la solution de ce problème de contrôle.

Contrôler le comportement d'un système peut être implanté par la structuration de modules selon des liens d'organisation qui expriment sur quoi se focalise le système et par la définition de séquences d'interactions entre eux. Ce contrôle a besoin d'être modifié à chaque fois que le fonctionnement global du système change selon le développement de la scène et selon les buts globaux à satisfaire. Chaque module doit être capable de définir ses propres buts pour réagir au développement de la scène et de planifier les interactions avec les autres modules. Le succès de la coopération entre ces agents de base pour la satisfaction de tâches globales telles que la reconnaissance d'objets est fortement dépendante des Protocoles d'Interaction qui peuvent être établis dans la société. Les Protocoles d'Interaction expriment, prédisent et vérifient les boucles qui implémentent des modes de traitement de haut en bas et du bas vers le haut selon le niveau de représentation des agents qui appartiennent à la même focalisation. En gardant ceci en vue, il est nécessaire que :

- les agents soient capable d'envoyer des messages et de les recevoir (*requête*, *réponse* ou *information*). Par conséquent on utilise le Langage de Communication présenté à la section 2.2,
- l'intention de l'émetteur et le comportement attendu du récepteur soit rendu le plus explicite possible. Pour cela on utilise le Langage Multi-Agents introduit à la section 2.3,
- le flux d'interaction entre agent soit amélioré, contrôlé et réduit au minimum nécessaire. On utilise donc les avantages du Protocole d'Interaction mentionnés à la section 2.4 pour implanter le contrôle global du système en termes de ses comportements possibles et souhaités,
- les agents partagent un Langage du Domaine commun. Ils doivent être capables de se comprendre. C'est évidement spécifique de chaque domaine d'application (comme la vision par ordinateur).

On pourra se reporter à [8] pour une étude sur deux cas de l'utilisation de protocoles d'interaction pour contrôler des systèmes de vision robotique.

La section suivante montre comment notre simulation de plate-forme de gestion de réseau pourrait être utilisée comme un système multi-agents générique pour la supervision de réseau.

3 Illustration en Gestion de Réseau de Télécommunications

Compte tenu de l'hétérogénéité des systèmes de gestion de réseau, l'interaction entre systèmes se fait en suivant les recommandations d'organismes de standardisation. Les modes d'interactions

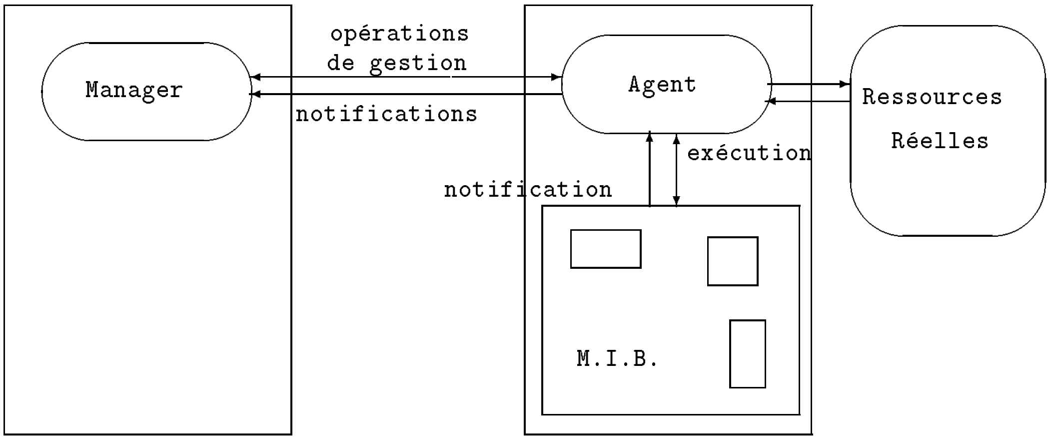

Figure 2: Le modèle Manager-Agent

présentés par la suite ont été définies par le groupe OSI (Open system Interconnexion) de l'ISO [13].

Dans le cadre de ses normes, une communication entre deux systèmes se fait par le biais d'une *association*. Cette association distingue deux rôles assurés par chacun des systèmes mis en relation. Un rôle manager et un rôle agent. Ils sont représentées sur la figure 2. Le rôle manager consiste à envoyer des requètes, à en recevoir les réponses ainsi que de recevoir des notifications. Le rôle agent consiste à recevoir les requêtes, les traiter, envoyer leur réponses ainsi que d'envoyer des notifications. Les notifications sont des messages émis par le système ayant le rôle d'agent qui correspondent à l'évolution de l'état des ressources internes que ce système gère. Cependant, pour recevoir une notification, le manager doit signaler à l'agent qu'il est intéressé par celle-ci en lui demandant de créer un filtre discriminateur. L'usage de ces filtres sera présenté lors de la présentation des protocoles d'interaction internes à l'agent, à propos de l'émission d'une notification (cf fig X).

Les spécificités des interactions entre systèmes de télécommunications dans le cadre de la supervision de réseaux étant maintenant rappelées, nous allons voir ce que pourrait apporter les outils précédemment décrits au niveau du *contrôle social* entre agents du réseau. Par ailleurs, la décomposition d'un système agent en sous-agents peut s'avérer intéressante dans le cadre de la supervision de réseau. En effet si la distribution physiques des applications de gestion d'un réseau semble évidente il est aussi intéressant de concevoir un système de gestion comme un système distribué. Nous montrerons donc ensuite comment ces systèmes peuvent se décomposer eux-mêmes en sous-agents, et voir, là encore, ce que pourrait apporter les outils précédemment décrits au niveau du *contrôle individuel* d'un agent ainsi décomposé.

3.1 Interactions et contrôle social entre agents

Dans le même esprit que celui qui a animé la réalisation des systèmes de vision précédemment décrits, le langage d'interaction et les protocoles peuvent être, de manière assez évidente, appliqués pour assurer le contrôle social entre agents situés sur le réseau. A ce niveau, il convient d'être prudent et de tenir compte des normes et outils existant dans le domaine de Télécommunications. Les recommandations ISO introduisent déjà certaines bases pour un tel langage d'interaction, et de fait, il existe certaines ressemblances entre les outils du LIFIA par rapport aux outils comme GDMO et CMIS.

GDMO (Norme ISO - Guideline for the Definition of Managed Objects), formalisme orienté objet de description, permet de décrire les informations de gestion en terme d'*objets gérés*, un objet géré étant défini comme une abstraction sur une ou plusieurs ressources physiques ou logiques du système géré.

Par ailleurs, la communication de gestion entre un gestionnaire et un agent s'effectue grâce aux services CMIS (Common Management Information Service). Ces éléments de service offrent au processus gestionnaire de réaliser sur un ou plusieurs objets gérés des opérations de gestion (création et destruction d'objets, consultation et modification d'attributs et déclenchement d'actions) et de rapporter des notifications.

Nous présentons tout d'abord GDMO et CMIS, en montrant les ressemblances avec le langage d'interaction développé au LIFIA, langage développé hors domaine d'application, puis nous envisageons comment ces différents outils peuvent se fondre l'un avec l'autre, dans l'attente de travaux et de résultats à venir et qui iraient dans le sens d'une possible unification.

3.1.1 La norme GDMO (Guideline for the Definition of Managed Objects)

GDMO est un formalisme orienté objet de description des ressources d'un réseau. Chaque ressource est considérée comme étant un *objet géré*. Les objets de même nature (i.e. ayant les mêmes propriétés) sont des instances d'une même classe d'objets. Ces classes sont regroupées dans un arbre d'héritage.

Une classe d'objets est un *gabarit*, c'est-à-dire une structure prédéfinie, représentée principalement par ses :

- attributs
- actions : opérations effectuées sur les attributs
- notifications : messages envoyés par un objet pour signaler un événement
- une description informelle de son comportement

Afin de rendre possible des requêtes sur un ensemble d'objets, les noms des objets sont organisés hiérarchiquement dans un arbre dit de nommage. Ainsi un objet est nommé de façon unique. L'arbre est défini par la relation de contenance, qui exprime le fait qu'un objet géré peut contenir d'autres objets, de la même classe ou d'une classe différente. Elle permet de modéliser des relations physiques (assemblage, composant...) ou des relations organisationnelles (répertoire, fichier, enregistrement, champ...) entre des ressources représentées par les objets.

L'ensemble conceptuel des objets d'un système géré constitue, par définition, la Base d'Informations de Gestion (en Anglais la MIB : *Management Information Basis*).

3.1.2 La norme CMIS (Common Management Information Service)

CMIS permet le dialogue entre deux entités du réseau en fournissant un ensemble de primitives qui sont utilisées pour créer et envoyer des requêtes ou des rapports d'événements. Les principales primitives sont :

- `m-get` : consulte la valeur de l'attribut d'un objet
- `m-set` : modifie la valeur de l'attribut d'un objet
- `m-create` : crée un objet avec ses valeurs par défaut
- `m-delete` : supprime un objet
- `m-action` : déclenche une action sur l'objet désigné
- `m-event-report` : émet un rapport d'événement (structure qui permet à un agent d'envoyer des notifications comme la création d'un objet, le dépassement d'un seuil...)

Les primitives de requête, en plus de la désignation de l'objet, peuvent avoir les attributs suivants, qui permettent la sélection d'un ensemble d'objets grâce à l'arbre de nommage :

- Le champ d'action (de communication) : il détermine la profondeur du sous-arbre de nommage sur lequel s'applique la requête.
- Le filtre : il permet de sélectionner un sous-ensemble du sous-arbre précédent en définissant des critères sur les attributs.

3.1.3 Perspectives

Les primitives exposées ici correspondent assez bien aux différentes valeurs possibles du champ `<type>` de notre langage multi-agents, puisqu'on retrouve les notions de requête, de réponse et d'information. Quant à la notion de présentation, elle est abordée dans d'autres normes, notamment ACSE, que nous ne présenterons pas ici. Etant donné d'autre part que la partie purement `<communication>` est également prise en charge par les couches inférieures de l'ISO, et sachant que la partie `<application>` est formalisée par GDMO, il ne nous resterait plus qu'à ajouter les champs `<nature>` et `<force>` pour obtenir l'équivalent de notre Langage d'Interaction.

Du point de vue des protocoles d'interaction, les agents peuvent communiquer à l'aide du langage d'interaction présenté précédemment, associé à des protocoles d'interaction qui sont utilisés pour contrôler le flux et le séquencement des messages. Chaque cas d'interaction entre les agents et le superviseur est représenté par un protocole de base qui lui est associé. Dans une approche classique, le protocole correspond effectivement au séquencement attendu de l'exécution des actions. Un cas dit dégradé, où des possibilités supplémentaires de transitions pourraient être ajoutées à ces protocoles de base, et pour lesquelles le choix serait laissé aux agents de les exécuter ou non, devrait permettre une plus grande adaptabilité et une plus grande autonomie des agents et un possible allégement des tâches des superviseurs.

3.2 Interactions et contrôle individuel d'un agent

Nous avons présenté une utilisation somme toute évidente du langage d'interaction et des protocoles d'interaction au niveau du contrôle dit social entre les agents du réseau. Un tel langage et de tels outils peuvent également être utilisés pour le contrôle individuel d'un agent du réseau.

3.2.1 Décomposition d'un agent en sous-agents

Un agent peut lui-même se décomposer en plusieurs sous-agents qui réagissent selon les différents rôles qu'il endosse. La figure 3 donne le détail de ces différents cas d'interactions entre les quatre sous-agents d'un agent que sont :

- le gestionnaire de communication GC, qui reçoit et envoie les messages vers les autres agents du réseau de télécommunication.
- le gestionnaire d'objets GO, qui met à jour les données et interprète les requêtes.
- le gestionnaire d'événement GE, qui détermine si les modifications amenées aux objets doivent donner lieu à un envoi de rapport d'événement. Il effectue des tests sur les informations reçues et décide si elles sont intéressantes ou pas et pour qui.
- l'application de gestion AP, décide du comportement de l'agent selon les événements reçus. Elle est spécifique aux ressources réelles, tels les équipements, qui luis sont connectés. Elle peut entre autre composer des requêtes et les envoyer aux autres agents par l'intermédiaire des gestionnaires de communication.

Notons que cette décomposition en sous-agents est d'autant moins artificielle que dans certaines plates-formes réelles d'administration de réseau, on retrouve effectivement une décomposition

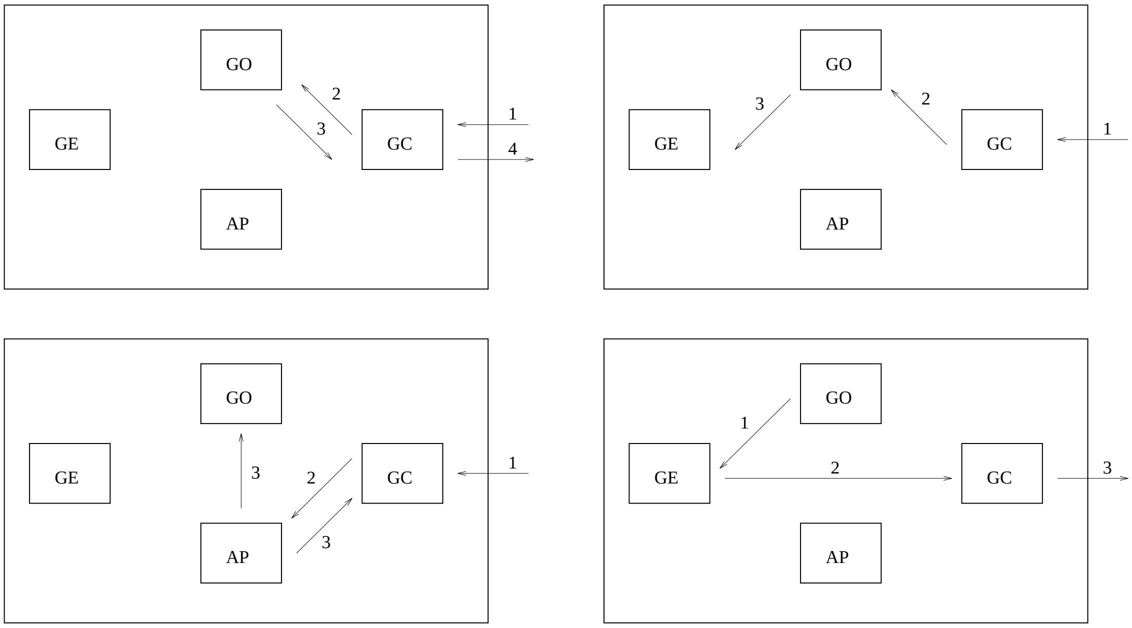

Figure 3: Les différents protocoles d'interaction primitifs entre sous-agents

des fonctions génériques d'une application de gestion, en agents, dits fonctionnels. Ces agents fonctionnels peuvent regrouper, pour plusieurs applications de gestion, des gestionnaires d'objets, des gestionnaires d'événements, ou autres services communs.

3.2.2 Les différents rôles de l'agent

Dans le cadre d'une application de supervision, le superviseur est manager vis-à-vis d'autres systèmes de gestion, afin de collecter de manière globale des rapports d'événements et d'envoyer des requêtes d'interrogation pour compléter ses informations. Dans la figure 3, les flèches des schémas représentent le flux de données, chaque message est dénoté par son numéro d'arrivée dans l'ordre chronologique. Les quatre schémas représentent les quatre protocoles d'interaction primitifs entre les différents sous-agents. Les différents rôles de l'agent ainsi dénotés consistent à (cf. figure 3 de haut en bas et de gauche à droite) :

- (de superviseur vers agent) traiter une requête du superviseur, requête simple, dans le cas d'une interrogation sur un attribut d'objet,
- (de superviseur vers agent) traiter une requête du superviseur, requête plus complexe, dans le cas d'une demande d'abonnement à un rapport d'événement induisant la création de filtres discriminateurs pris en charge par le gestionnaire d'événements,
- (de équipement vers agent) réceptionner un rapport d'événement venant d'un autre agent, l'agent devenant lui-même manager, ou d'une notificiation venant d'un équipement,
- (de agent vers superviseur) diffuser une notification à l'attention du superviseur.

3.2.3 Perspectives

Plus généralement, les sous-agents peuvent communiquer à l'aide du langage d'interaction présenté précédemment, associé à des protocoles d'interaction qui sont utilisés pour contrôler le flux et le séquencement des messages. Chaque cas d'interaction entre les sous-agents est représenté par

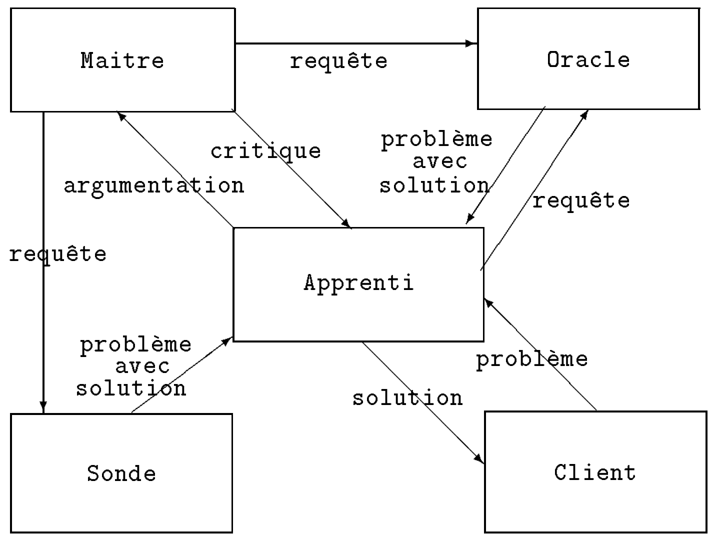

Figure 4: Un protocole multi-agents, la pentade MOSCA

un protocole de base qui lui est associé. Dans une approche classique, le protocole correspond effectivement au séquencement attendu de l'exécution des actions. Un cas dit dégradé, où des possibilités supplémentaires de transitions pourraient être ajoutées à ces protocoles de base, et pour lesquelles le choix serait laissé aux sous-agents de les exécuter ou non, devrait permettre une plus grande adaptabilité et une plus grande autonomie des sous-agents et des agents eux-mêmes.

C'est dans ce cadre d'utilisation des protocoles d'interaction au niveau du contrôle individuel d'un agent que peut être envisagé le protocole MOSCA du LIRMM. Dans le protocole multi-agents d'apprentissage MOSCA [14], une situation typique d'apprentissage est modélisée (figure 4), et les différents types de messages que peuvent s'envoyer les agents correspondent aux interactions entre les différents éléments intervenant dans un processus d'apprentissage automatique. Appliqué dans le domaine des télécommunications, le protocole MOSCA pourrait ainsi être utilisé par le LIRMM pour implanter et optimiser le contrôle des échanges d'information qui ont lieu dans au sein des agents.

4 Conclusion : Vers une programmation récursive orientée agents

Dans cet article, nous avons présenté un "langage d'interaction", inspiré de la théorie des actes du langage et préconisons son utilisation dans le cadre de la supervision de réseaux de télécommunications.

Du point de vue du domaine d'application, nous venons de voir qu'un certain nombre d'interactions ont lieu au sein même des agents en télécommunications et que ces interactions apparaissent entre quatre entités clairement définies dans cet agent. Cette décomposition des agents et cette mise en évidence des activités de communications qui sont à la base de son comportement, suggèrent d'appliquer les notions de contrôle des interactions dans une société d'agents dans le contexte du groupe de sous-agents que forme l'agent. C'est donc l'utilisation uniforme de mêmes outils au niveau des agents et des plates-formes d'administration de réseau qui est ici mise en avant, pour un développement plus intégré et plus aisé des applications en gestion de réseau dans le cadre de plates-formes qui fournissent un ensemble de services fonctionnels à utiliser par lesdites applications de gestion. Notre proposition n'est néanmoins pas tout à fait nouvelle dans le sens où il semble qu'elle rejoigne l'approche suivie par Bull dans sa plate-forme d'administration de réseau Bull-ISM (Integrated System Management) qui généralise l'usage des protocoles de communication ISO aussi bien à l'intérieur de la plate-forme que pour la communication entre plates-formes.

Du point de vue du langage d'interaction, cela signifie que le langage peut servir non seulement à faire interagir différents agents et superviseur d'un réseau de télécommunications, mais également à modéliser les différentes interactions entre rôles primitifs qu'un agent du réseau peut assurer. Aller ainsi d'une interaction entre agents vers des interactions entre primitives d'un agent, en utilisant un même formalisme et un même outil, nous semble constituer un premier pas essentiel vers des langages orientés agent qui soient de réels langages de programmation.

5 Remerciements

Le travail du LIFIA et du LIRMM sur les systèmes multi-agents bénéficie du soutien du GDR-PRC IA. Dans le cadre de l'application au domaine des Télécommunications, ce travail béneficie également du soutien du CNET au titre du marché 93 1B 142 entre le CNET et le LIRMM. Dans le cadre des applications au domaine de la Télématique, le travail du LIFIA bénéficie du soutien de l'Union Européenne à travers sa participation dans l'étude de faisabilité GEOMED (Telematics - Engineering Project 174). Les auteurs tiennent également à remercier Michel Plu, de France Telecom - CNET (Lannion A/EIA/AIA), pour ses nombreuses remarques et corrections apportées aux premières versions de ce papier.

References

[1] T. Ballmer and W. Brennenstuhl Speech Act Classification : A study in the lexical analysis of speech activity verbs. Springer Verlag, 1981.

[2] Jean Pierre Briot. From objects to actors: Study of a limited symbiosis in smalltalk-80. Research Report RXF 88-58, LITP, Univ Paris 6, 1988.

[3] B. Burmeister and K. Sundermeyer. Cooperative problem solving guided by intentions and perception. In Werner and Yves Demazeau, editors, *Decentralized AI III.* Elsevier Science, 1992.

[4] B. Burmeister, A. Haddadi and K. Sundermeyer. Generic configurable cooperation protocols for multi-agent systems. In Proceedings of the fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW '93, Neuchatel, CH, August 1993.

[5] J. Campbell and M. D'Iverno Knowledge interchange protocol In Decentralized AI I, Demazeau and Muller eds, Elsevier Science Publishers B.B., 1990.

[6] M. Chang and C. Woo SANP: a communication level protocol for negotiations In Decentralized AI III, Werner and Demazeau eds, Elsevier Science Publishers B.B., 1992.

[7] S. Berthet, Yves Demazeau and O. Boissier. Knowing each other Better. In *Proceedings of the 11th International Workshop on Distributed Artificial Intelligence*, IWDAI '92, Glen Arbor.

[8] Yves Demazeau, Olivier Boissier, and Jean-Luc Koning. Using interaction protocols to control vision systems. In *International Conference on Systems, Man and Cybernetics*, San Antonio, Octobre 1994. IEEE.

[9] Yves Demazeau. From Interaction to Collective Behavior in Agent-Based Systems. European Conference on Cognitive Science '95, Saint-Malo, Avril, 1995.

[10] E. Durfee and T. Montgomery A hierarchical protocol for coordinating multiagent behavior. In *Proceedings of the AAAI '90 Conference*, AAAI, 1990.

[11] G. Gaspar. Communication and belief changes in a society of agents: towards a formal model of autonomous agent. In Decentralized AI II, Demazeau and Muller eds, Elsevier Science Publishers B.B., 1991.

[12] M. R. Genesereth and S. P. Ketchpel. Software agents. *ACM*, Juillet 1994.

[13] Lecerf and Chaumel. *Les normes de gestion de réseau à l'ISO*. Masson, 1992.

[14] J. Quinqueton, P. Reitz and J. Sallantin. Les shéma mentaux, un cadre conceptuel pour l'apprentissage à partir d'exemples. In J.P. Laurent, editor, Reconnaissance des Formes et Intelligence Artificielle, Lyon, novembre 1991.

[15] P. Reitz and J. Quinqueton. Mosca: a multi agents learning framework. In Proceedings of ECML93, poster session, 1993.

[16] J. Searle Speach Acts. Cambridge University Press, 1969.

[17] Yoav Shoham. Agent oriented programming. Research Report STAN-CS-90-1335, Stanford University, Octobre 1990.

[18] R. Smith. The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Transactions on Computers, Vol 29, IEEE, 1980.