Philippe POPULAIRE, Yves DEMAZEAU, Olivier BOISSIER, Jaime SICHMAN, "Description et implémentation de protocoles de communication en univers multi-agents", 1ères journées francopohones Intelligence Artificielle Distribuée et Systèmes multi-agents, Toulouse, Avril 1993

DESCRIPTION ET IMPLEMENTATION DE PROTOCOLES DE COMMUNICATION EN UNIVERS MULTI-AGENTS

Philippe POPULAIRE, Yves DEMAZEAU*
Olivier BOISSIER, Jaime SICHMAN¥

LIFIA/IMAG
INPG 46 avenue Félix Viallet 38031 Grenoble Cedex
Email : Yves.Demazeau, Olivier.Boissier, Jaime.Sichman@imag.fr Tel : 76 57 46 54

MOTS-CLES

Systèmes multi-agents, Protocoles de communication, Implémentation.

RESUME

Cet article présente un système pour la création et l'utilisation de protocoles de communication dans des systèmes multi-agents. Ces protocoles sont basés sur les notions d'actes de langage. Un langage général de description est proposé pour les exprimer. Un système est développé sur la base de ce langage pour la gestion de ces protocoles permettant : leur distribution au sein de chacun des agents, leur évolution dynamique dans le cours de la résolution et la coexistence de plusieurs versions d'un même protocole à un moment donné dans le système. L'expression de différents protocoles de communication issus de la littérature des systèmes d'Intelligence Artificielle Distribuée a constitué une première phase de validation de la généralité du langage de description développé. L'implantation et la mise en oeuvre du système de gestion offre de nouvelles perspectives sur les plate-formes pour le développement de systèmes multi-agents.

---

* LIFIA/IMAG/CNRS

¥ En détachement de l'Ecole Polytechnique de l'Université de São Paulo, Brésil, financé par la FAPESP, bourse 91/1943-5

1. INTRODUCTION

Dans les systèmes multi-agents, la communication est un des éléments importants pour assurer la cohérence de résolutions entreprises localement par chacun des agents. Des protocoles de communication sont utilisés dans ces systèmes afin de contrôler et de structurer les échanges d'informations entre agents de façon à réduire le temps de communication et la quantité d'informations échangées. Ils imposent des règles à la communication et constituent donc une loi de comportement pour les agents.

Le développement de protocoles est lié à l'expression de stratégies de résolution de problèmes découlant de l'existence de plusieurs agents dans un même système : résolution de conflits (Smith,83) (Chang,91), apprentissage coopératif (Sian,91), introduction d'un nouvel agent dans un système (Berthet,92), partage de tâches (Kreifelts,91). Pour chacun de ces protocoles, un formalisme particulier est développé. Il dépend du type et du langage de communication inscrits dans le système. L'objectif de cet article est de présenter un langage unique pour l'expression de ces divers protocoles. Les diverses fonctionalités pour l'interprétation de ce langage sont regroupées dans un système assurant la consultation et la gestion des protocoles développés. Ce système est distribué dans chacun des agents.

Dans une première section, nous définissons le cadre dans lequel nous envisageons les notions de protocole et de communication dans les systèmes multi-agents. La deuxième section est dédiée à la présentation du langage de description de protocoles que nous avons conçu en l'illustrant avec un protocole issu de la littérature IAD. Nous présenterons ensuite le système général que nous avons conçu et implanté pour la création et l'utilisation des protocoles définis à l'aide de ce langage. Son implantation sera présentée dans la quatrième section. Nous conclurons enfin en donnant les perspectives de notre étude.

2. PROTOCOLES DE COMMUNICATION

2.1. Actes de langage

Du point de vue d'un agent, nous considérons la communication comme un cas particulier d'action sur le monde extérieur, liée à une intention de l'agent ayant émis le message. Nous rejoignons ainsi le courant qui, actuellement en IAD, s'inspire des études sur les *actes de langage* (Searle,69) pour la formulation de la communication dans les systèmes.

Afin de situer le contexte de la communication, le destinataire du message doit pouvoir reconnaître l'intention de l'émetteur. Il doit ainsi pouvoir séparer, dans le message, la partie intentionnelle du contenu purement propositionnel. Par exemple, le contenu propositionnel d'une phrase du type "je serai là à 5 heures" est l'information "il sera présent à 5 heures", la partie intentionnelle est, par exemple, que le locuteur s'engage à être là.

Nous choisissons ainsi d'exprimer une interaction comme combinaison du contenu propositionnel et d'une *force d'élocution* exprimant le contenu intentionnel de l'acte de langage. Dans une forme schématique :

acte de langage = force d'élocution + contenu propositionnel.

Diverses classifications des actes de langage selon leur force d'élocution ont été proposées sur la base de celle proposée par Searle (Searle,69) : classification selon vingt et une forces (Campbell,90) dans laquelle nous trouvons, par exemple, l'expression des intentions relatives à la recherche d'informations (*Information seeking*), à la persuasion (*Persuading*), à l'avertissement (*Warning*), au conseil (*Advising*), au marchandage (*Bargaining*), au commandement (*Commanding*) et à l'expression (*Expressing*). Elles permettent d'élargir, les types d'interaction les plus courants dans un système multi-agents qui correspondent à une proposition d'informations (*Inform*), à une requête d'informations ou d'actions (*Request*) ou à une réponse à une requête (*Answer*).

2.2. Protocoles de communication

De nombreux protocoles de communication sont présentés dans la littérature des systèmes multi-agents. Ainsi par exemple, dans le système de Gaspar (Gaspar,91) un protocole simple exprimé par quatre types de messages correspond aux modes de communication entre les agents.

Dans le système développé par Sian (Sian,91) le protocole met en oeuvre une méthode d'apprentissage pour un système multi-agents. A partir d'informations locales, chaque agent déduit de nouvelles hypothèses et maintient leur degré de confiance par interaction avec les autres agents via un tableau noir. Chaque agent comporte un module d'apprentissage réalisant l'apprentissage à partir des interactions, un module mémorisant l'expérience et un module de traitement utilisant cette expérience pour génèrer les actions.

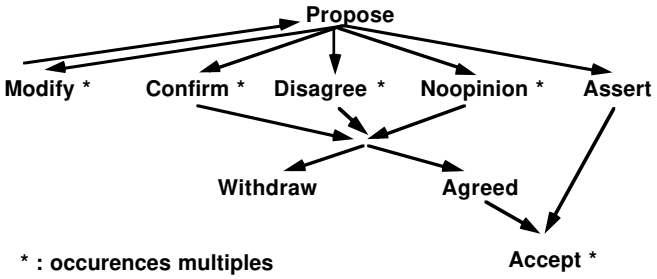


Fig.1. Protocole d'échange de connaissances (Sian,91).

Neuf primitives expriment les actes de langage pour l'émission d'une hypothèse non modifiable (*Assert*), la proposition d'une nouvelle hypothèse (*Propose*), la modification d'une hypothèse émise précédemment (*Modify*), l'acceptation d'une hypothèse (*Agree*), le désaccord vis-à-vis d'une hypothèse (*Disagree*), l'absence d'opinion vis-à-vis d'une hypothèse (*Noopinion*), la confirmation de la validité d'une hypothèse (*Confirm*), le retrait d'une hypothèse précédemment proposée (*Withdraw*). Le protocole de communica-

tion (cf. Fig. 1.) réduit les échanges possibles entre les agents en définissant les séquences possibles d'utilisation de ces primitives. Il consiste en trois phases : génération d'hypothèses, coopération afin de s'accorder sur les hypothèses et intégration des informations résultant des échanges.

Des protocoles plus complexes ont été élaborés par Chang et Woo (Chang,91), par exemple, qui se sont basés sur les résultats d'études linguistiques sur l'expression des conflits dans les sociétés humaines. Un protocole constitué de six étapes est ainsi proposé. Les diverses intentions de communication sont exprimées au sein de ce protocole.

Comme nous le voyons aucune normalisation de ces protocoles n'est actuellement disponible. Couvrant divers aspects des échanges qui peuvent se dérouler dans un système multi-agents, le regroupement de ces protocoles en une structure unique est d'un immense intérêt.

3. LANGAGE DE DESCRIPTION DE PROTOCOLES DE COMMUNICATION

Ayant définit le contexte dans lequel nous nous plaçons, nous présentons, dans un premier temps, le langage de description que nous avons défini, pour établir une base commune de description de protocoles d'Intelligence Artificielle Distribuée. Nous définissons un protocole de communication sous la forme d'une suite d'actes de langage plus ou moins spécifiés. Le déroulement possible de la communication est ainsi fixé tout en laissant à l'agent la liberté de définir les éléments non spécifiés en tenant compte de ses intérêts locaux ou d'autres éléments exprimés par exemple dans l'organisation qui lui assigne un rôle de communication (Boissier,93).

3.1. Protocoles

L'expression du protocole se traduit ainsi par une succession d'étapes élémentaires de traitement local mis en oeuvre au sein des agents, s'intercallant entre toute réception et émission d'interactions. Chacune de ces étapes est représentée par un *état*. Une interaction provoquant le passage d'une étape à une autre est représentée par une *transition*. Les états sont regroupés en *protocoles* permettant d'exprimer des enchainements relatif à un type de comportement de communication dans le système (ex : protocole de doute, protocole de vérification). Les protocoles sont eux-même regroupés en *lois*. Ces dernières expriment des comportements de communication relatifs à des résolutions particulières (résolution de conflit, présentation, apprentissage). Les transitions possibles associées à un état sont celles issues de cet état vers les autres états. Elles constituent l'ensemble des choix offerts à l'agent à chaque étape du protocole pour la poursuite des échanges.

Nous donnons une description du langage selon la norme de Backus-Naur. Les symboles terminaux sont écrits en caractères gras. Les variables syntaxiques sont en caractères normaux entre les symboles < et >.

<table_de_lois> ::= **version** <date> <table_de_lois> |
law <nom> **{** <lprotocoles> **}** <table_de_lois> |
law <nom> **{** <lprotocoles> **}**
<date> ::= <jour> / <mois> / <année>
<lprotocoles> ::= **protocol** <nom> **{**<létats>**}** <lprotocoles> |
protocol <nom> **{**<létats>**}**
<létats> ::= **state** <nom> **{** <ltransitions> **}** <létats> |
state <nom> **{** <ltransitions> **}**

Les variables <jour>, <mois> et <année> sont des entiers indiquant la date de création des lois. Leur existence se justifie par la coexistence possible, dans le système, de deux ou plusieurs versions différentes de la même loi.

3.2. Transitions

Une transition est la spécification de l'acte de langage (message) à laquelle est ajoutée, optionnellement, une condition portant sur les états internes de l'agent. Cette spécification est exprimée par la combinaison de quatre champs relatifs, respectivement, au type du message, au routage, à la force d'élocution et au contenu. La condition est une expression booléenne permettant de déterminer la possibilité d'envoyer le message spécifié dans la transition.

Le type de message exprime le comportement de communication à utiliser : requête (**request**), réponse (**answer**), information gratuite (**inform**).

Le routage indique si le message est ciblé (**you**), diffusé à tous (**broadcast**) ou seulement à un groupe d'agents (groupe_d_agents). La définition de ce dernier terme est dépendante de l'application. Un *appel d'offre* (cf. protocoles du type Contract Net (Smith,83)) devient dans ce formalisme une requête envoyée en diffusion : (request)(broadcast).

La force d'élocution est, comme nous l'avons précisé ci-dessus, l'intention du message. Elle peut être exprimée par l'un des vingt et un tons de Campbell et D'Inverno (Campbell,90) ou par toute autre sélection de tons compris par les agents.

Le contenu correspond au contenu propositionnel du message. Nous détaillons dans la section suivante son expression.

L'état découlant de l'exécution de la transition, étape suivante, est spécifié par la combinaison des noms de loi, de protocole et d'état. L'absence de l'un de ces noms conduit à l'utilisation de sa valeur courante.

L'expression des transitions est la suivante :

<ltransitions> ::= **[**<transition>**];** | **[** <transition> **]** | <ltransitions>
<transition> ::= **if (**<condition>**)**<message> | <message> | **end**

<message> ::= <lchamps> **->** <étape>
<lchamps> ::= **(**<type>**)(**<routage>**)(**<force>**)(**<contenu>**)**
<type> ::= **inform** | **request** | **answer**
<routage> ::= **you** | **broadcast** | <groupe_d_agents>
<étape> ::= **law** <nom> **protocol** <nom> **state** <nom> |
protocol <nom> **state** <nom> |
state <nom>

Les variables <nom>, <force> et <groupe_d_agents> sont des chaines de caractères définies en fonction des protocoles et de l'application. La <condition> et le <contenu> sont définis en fonction de l'application. Nous les expliciterons dans le cadre de la section suivante.

Le langage ne permet d'exprimer ni les comportements dans le cas d'absence de réponse ni à partir de quel moment on considère qu'il n'y a pas de réponse. Nous considérons en effet que le comportement à adopter est fonction de l'agent, de ses contraintes, de la nature de son interlocuteur (ex: sa vitesse de traitement), et de l'importance accordée à la communication.

3.3. Expression du contenu et des conditions

Bien que la définition des champs <contenu> et <condition> soit proche de l'application envisagée, nous ajoutons au langage précédent un ensemble de termes de base pour leur expression. La <condition> est une expression booléenne dont les termes sont des mots clefs faisant références aux symboles relatifs à l'application, manipulés par l'agent. La syntaxe du contenu est la suivante :

<contenu> ::= <nom> |
<nom> **=** <nom> |
<contenu> **,** <contenu>

Les symboles **=** et **,** définissent respectivement les opérateurs d'affectation et de juxta-position. Les valeurs de la variable <nom> sont de deux types selon qu'elles sont ou non dépendantes de l'application. L'écriture de différents protocoles de la littérature (Populaire,92) dans notre langage nous a conduit à isoler des mots-clefs introduisant les termes spécifiques à l'application ou à la fonction du protocole. Ils expriment : le sujet principal sur lequel porte un échange d'information (**matter**), le sujet secondaire pouvant remplacer le sujet principal en cours d'échange (**help**), la raison d'un message ou d'une action (**reason**), le résultat d'une phase d'échange entre les agents (**result**). Diverses applications en cours de développement au sein du laboratoire ont permis de compléter ce

langage en introduisant différents mots-clefs particuliers à des échanges d'information dans un système intégré de vision par exemple (Boissier,93).

3.4. Exemple d'utilisation

Afin d'illustrer ce langage, nous exprimons le protocole de Sian présenté dans la première section. Les actes de langage exprimés dans ce protocole sont reformulés dans notre langage par la combinaison du type, des forces d'élocutions et du contenu. Le langage de définition du contenu est relatif à l'échange des règles de déduction de nouvelles hypothèses entre les agents. Il utilise les mots clefs **rule** et **new_rule**, variables instanciées par les règles échangées. L'utilisation du tableau noir dans le système de Sian, accessible à tous les agents se traduit de manière explicite par la valeur **broadcast** dans le champ routage.

```
law Sian
  { protocol proposal
    { state init
      { (inform)(broadcast)(information_seeking)(matter=propose,rule)
          -> state opinion]; }
    state opinion
      { [(inform)(broadcast)(information_checking)(matter=modify_into,
          rule, new_rule)-> state init]|
        [(inform)(broadcast)(expressing)(matter=confirm,rule)
          -> state decision]|
        [(inform)(broadcast)(expressing)(matter=disagree,rule)
          -> state decision]|
        [(inform)(broadcast)(informing)(matter=noopinion,rule)
          -> state decision]; }
    state decision
      { [(inform)(broadcast)(warning)(matter=withdraw,rule)
          -> state end]|
        [(inform)(broadcast)(expressing)(matter=agree,rule)
          -> state agreement]; }
    state agreement
      { [(inform)(broadcast)(expressing)(matter=accept,rule)
          -> state end]; }
    state end
      { [end]; } }
  protocol assertion
    { state init
```

```
        {[(inform)(broadcast)(commanding)(matter=assert,rule)
              -> protocol proposal state agreement]; }
    }
}
```

Parallèlement à cette définition formelle du langage de description d'un protocole, nous avons défini une représentation graphique sous la forme d'un graphe d'états normalisé. Dans le cas du protocole précédent, on aura ainsi, pour un échange de connaissances entre un agent A et un agent B, le graphe normalisé suivant (cf. Fig. 2):

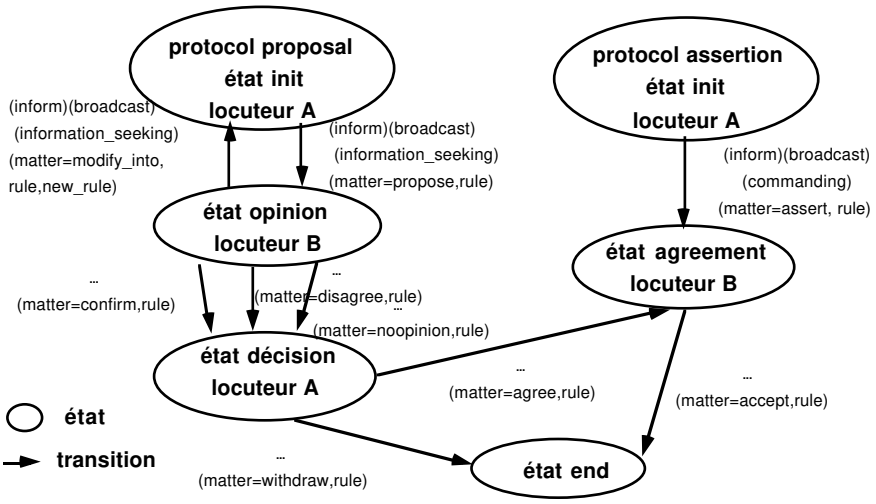


Fig. 2. Protocole de Sian, graphe d'état normalisé.

4. SYSTEME POUR LA GESTION ET LA DEFINITION DES PROTOCOLES DANS UN SYSTEME MULTI-AGENTS

La généralité de ce langage offre une base stable pour l'implantation de nombreux protocoles rencontrés dans la littérature (Populaire,92). Nous avons défini un système pour le développement et l'utilisation de ces différents protocoles s'insérant dans la plate-forme de développement actuellement en cours de réalisation au LIFIA.

4.1. Choix de réalisation

Afin de permettre que les protocoles de communication puissent s'appliquer et s'adapter à un sous-groupe d'agents, l'ensemble des protocoles est dupliqué dans chacun des agents (*localité* des protocoles). Afin de respecter le caractère intrinsèquement distribué découlant du choix précédent, nous avons préféré une gestion *décentralisée* (mise en oeuvre des protocoles au sein de chaque agent) à une gestion centralisée des protocoles. Un protocole peut avoir un rôle purement consultatif ou forcer l'agent à suivre les règles de communication qu'il préconise. On parle dans le premier cas de protocole *passif* et dans le second cas de protocole *actif*. Afin, d'une part, de réduire l'interaction entre le modèle d'agent et le système de gestion des protocoles et, d'autre part, de rendre la

couche de communication indépendante des protocoles, nous avons opté pour des protocoles passifs. Chacun des agents consulte le protocole et effectue le choix de transitions en respectant ou non les propositions indiquées par le protocole.

Les fonctions du système de gestion des protocoles permettent la lecture, la modification et la consultation des protocoles codées dans le langage décrit ci-dessus. Pour chacune de ces fonctions, des fonctions de traduction du langage dans celui qu'utilise l'agent sont développées selon les modèles d'agent utilisés. Des fonctions d'interprétation du langage utilisé pour l'expression des champs **condition** et **contenu** sont aussi ajoutées. Le choix des transitions proposées par le protocole, la définition du contenu de la transition choisie incombent à l'agent.

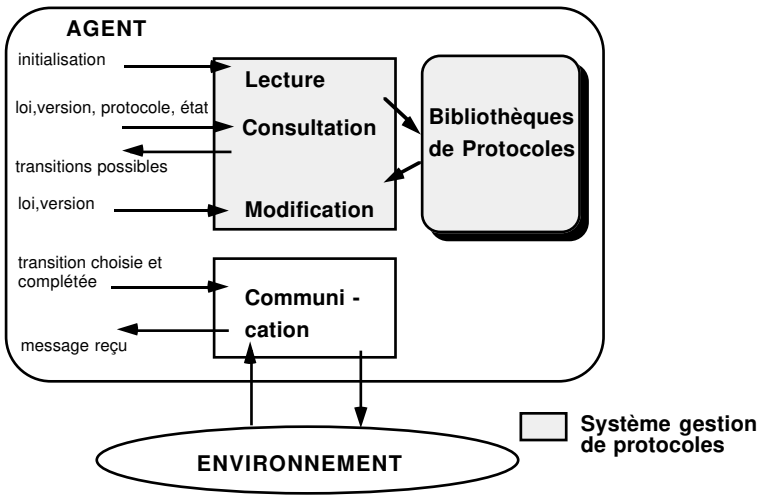


Fig.3. Architecture du système de gestion des protocoles de communication.

4.2. Lecture des lois

Utilisant des lois locales dont la gestion est décentralisée, la constitution initiale de la bibliothèque de lois au sein de chacun des agents (cf. Fig. 3) s'effectue par lecture d'une version globale de cette bibliothèque accessible à tous les agents. Une autre option, encore inexplorée, consiste en la demande par un nouvel agent introduit dans le système, d'une copie de la bibliothèque de protocoles à un agent existant.

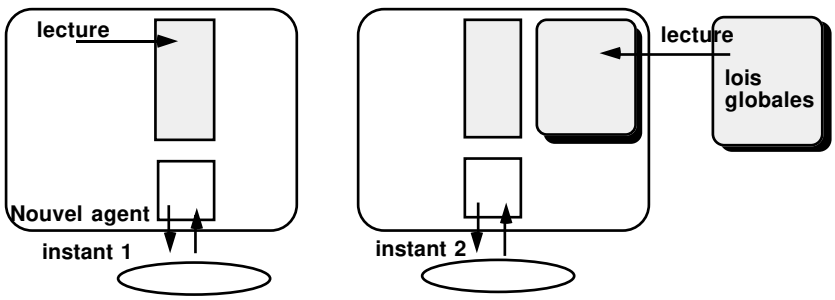
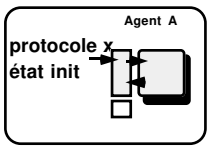


Fig.4. Initialisation des protocoles, transcription au niveau d'un agent

Un autre mode d'initialisation des lois locales, utilisé lorsqu'un agent omet la phase d'initialisation précédente, consiste en une lecture sélective des lois globales en fonction du protocole en cours d'utilisation. La transcription initiale de la totalité des lois globales n'est donc pas indispensable.

4.3. Consultation des lois

Lorsqu'il veut communiquer, l'agent consulte la bibliothèque de protocoles en spécifiant la loi qu'il veut utiliser (cf. Fig. 5.a). Il reçoit, en retour, l'ensemble des transitions possibles à partir de l'état initial du protocole initial de la loi (cf. Fig. 5.b). L'agent sélectionne une transition parmi celles possibles et complète les différents champs du message et l'envoie à son interlocuteur (cf. Fig. 5.c). Afin de permettre une mise en oeuvre simple et efficace du suivi d'un protocole entre des agents, nous avons choisi de communiquer dans tout message, l'étape suivante de la loi à laquelle conduit la transition choisie par l'émetteur. Lors de l'interprétation du message reçu, l'agent récepteur détermine les transitions permises pour continuer l'échange, à partir de l'étape transmise. Il consulte sa bibliothèque locale de protocoles (cf. Fig. 5.d), exécute les différentes étapes décrites ci-dessus et envoie le message choisi (cf. Fig. 5.e. et 5.f.). Le déroulement du protocole se poursuit ainsi jusqu'à sélection d'une transition **end** (fin de l'échange).



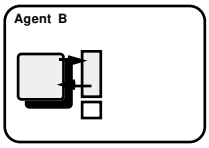
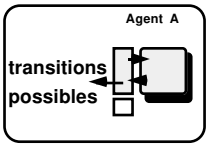


Fig. 5.a. consultation des protocoles



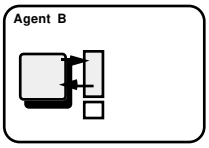


Fig. 5.b. résultats de la consultation

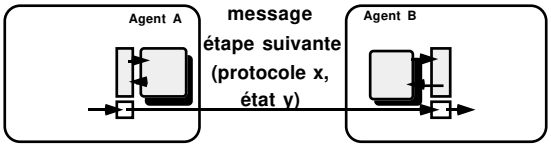
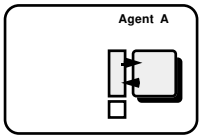


Fig. 5.c. construction du message et émission



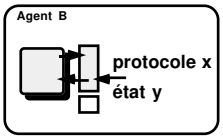
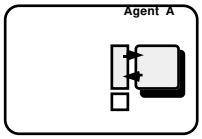


Fig. 5.d. consultation des protocoles



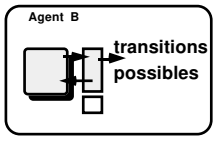


Fig. 5.e. résultats de la consultation

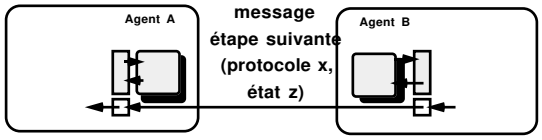


Fig. 5.e. construction du message et émission

4.4. Modification des lois

Une modification globale des protocoles nécessite, au préalable, une modification locale effectuée par un ou plusieurs agents et propagation de la nouvelle version. Une modification au niveau local porte sur la bibliothèque de protocoles d'un agent. Elle peut

être ensuite suivie d'une propagation de la modification vers tous les agents susceptibles d'utiliser cette nouvelle loi. La propagation est réalisée de deux manières possibles : elle est commandée par la demande explicite d'un agent aux agents concernés de remettre à jour leur bibliothèque locale ou par la constatation d'un "défaut de protocole". Dans ce dernier cas, l'agent auquel on s'est adressé n'a pas pu trouver le protocole nécessaire et répond par une demande de transmission de ce protocole. Afin de distinguer plusieurs versions successives d'un même protocole, une estampille est associée à chaque protocole. Actuellement elle se résume à la date de création.

5. IMPLEMENTATION

La définition du système complet nécessite le développement d'interfaces entre l'agent, le système de gestion des protocoles et l'environnement. Les différentes fonctions du système de gestion des protocoles sont écrites en C. Une structure arborescente organise les différentes lois et protocoles constituant la bibliothèque (cf. Fig. 6.)

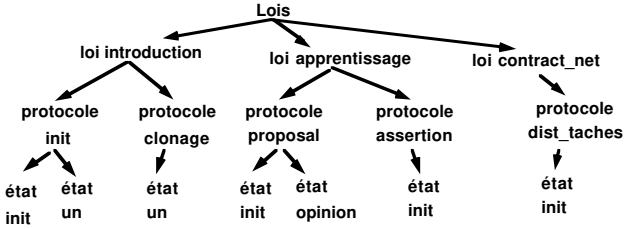


Fig. 6. bibliothèque de protocoles.

Le module de communication prend en charge tous les problèmes de mise en forme, envoi et réception de messages. Ce module est actuellement basé sur l'utilisation d'un système de résolution distribuée de problèmes orienté objets, DPSK+P (Sichman,92). Ce système repose sur l'échange d'objets entre applications réparties sur un réseau de machines. Il permet de travailler dans un environnement matériel hétérogène et ouvert. Nous nous affranchissons, ainsi, des problèmes de compatibilités matérielles. L'extension de l'application sans interruption de son fonctionnement est ainsi permise.

6. CONCLUSION

Nous avons conçu un langage de description de protocoles permettant de décrire plusieurs protocoles développés en Intelligence Artificielle Distribuée. Il ouvre la voie à une unification de protocoles de communication et à la création d'une bibliothèque de protocoles couvrant les différents besoins de communication pouvant exister dans un système multi-agents. Au cours de cette première phase de développement, nous avons établi une bibliothèque comprenant, actuellement, sept protocoles: CNET (Smith,83), SANP (Chang,91), protocole pour l'apprentissage (Sian,91), protocoles liés à l'introduction d'un nouvel agent dans un système (Berthet,92), protocole d'avertissement (Campbell,90). Il a pu aussi être testé dans le cadre du développement d'un système intégré de vision (Boissier,93). Notre objectif est, à présent, de mettre en oeuvre, tester et

affiner ces protocoles. Parallèlement à l'extention de cette bibliothèque, nous souhaitons essayer de dégager selon la même philosophie, des langages d'expression du contenu des échanges, qui comme nous l'avons dit, sont dépendants de l'application.

Un autre aspect intéressant dont ce travail n'a fait que fournir la base pour une étude plus approfondie, consiste en la modification des protocoles de communication par les agents eux-même en cours d'exécution du système.

7. BIBLIOGRAPHIE

Berthet S., Demazeau Y., Boissier O.,
Knowing each other better, 12th International Workshop on DAI, 1992

Boissier O.
Problème du contrôle dans un Système Intégré de Vision, Thèse de doctorat de l'INP Grenoble, Janvier 1993

Campbell J. and D'Inverno M.
Knowledge interchange protocols, Decentralized Artificial Intelligence, Elsevier Science Pub. B.V.North Holland, 1990.

Chang M.K., Woo C.C.,
SANP : A Communication Level Protocol for Negotiations, 3rd MAAMAW, Kaiserslautern, 1991

Gaspar G.,
Communication and belief changes in a society of agents. Towards a formal model of autonomous agents, Decentralized AI 2, Elsevier Sc. Pub.(North Holland), 1991

Kreifelts T., Von Martial F.,
A Negotiation Framework for Autonomous Agents, Decentralized AI 2, Elsevier Sc. Pub.(North Holland), 71-88, 1991

Populaire P.
Validation de protocoles de communication en systèmes multi-agents, Travail de fin d'étude, LIFIA-IMAG et Faculté Polytechnique de Mons, 1992

Searle J.R.
Speech Acts, Cambridge Univerity Press, 1969.

Sian S.S.
Adaptation Based on Cooperative Learning in Multi-Agent Systems, Decentralized Artificial Intelligence 2, Elsevier Science Pub. B.V. North Holland, 1991.

Sichman J.S., Cardozo E.,
A monitoring facility for an object-oriented distributed problem solving kernel, Représentations par objets: le point sur la recherche et les applications 1, 281-290, EC2, La Grande Motte, 1992

Smith R.G., Davis R.,
Negotiation as a Metaphor for Distributed Problem Solving, A.I. Jour., Vol 20, 1, 63-109, 1983