Un générateur de systèmes experts réparti est-il un système multi-agents ?

Wided Lejouad
Projet Secoia
INRIA/CERMICS Sophia-Antipolis
2004 route des Lucioles BP 93
F-06902 Sophia Antipolis Cedex (France)
E-mail : lejouad@sophia.inria.fr

Résumé

Un générateur de systèmes experts réparti (GSER) est un système modulaire et distribué sur plusieurs processeurs. L'échange d'informations entre ces modules est réalisé à travers un réseau informatique. Cette architecture offre au système la possibilité de réutiliser ses composants et les maintenir sans remettre en cause l'ensemble du système, d'où l'accroissement de son efficacité. Les composants d'un générateur de systèmes experts comme Smeci sont : le langage de schémas permettant de structurer la connaissance et de décrire son comportement, le langage des règles permettant de déduire de nouvelles connaissances, le moteur d'inférence et l'arbre d'états constituant la trace du raisonnement. Concevoir une architecture distribuée pour ce système a nécessité non seulement une étude de son découpage, une tâche importante qui prend en compte aussi bien la modularité du système que sa distribution, mais aussi une étude du protocole de communication nécessaire au dialogue entre composants.

Après les quelques précisions données sur la notion de GSE réparti, nous nous posons la question suivante : *pourrions-nous rattacher ce type d'application à l'une des disciplines d'Intelligence Artificielle (IA), en l'occurrence, l'IA Distribuée (IAD) et les Systèmes Multi-Agents (SMA) ?*

1 Introduction

Il est intéressant de passer en revue l'évolution des techniques d'intelligence artificielle [18], illustrée sur la figure 1, afin de permettre au lecteur de situer nos travaux par rapport à ceux abordés dans la littérature.

1.1 Intelligence artificielle classique

L'intelligence artificielle classique a été marquée par les systèmes à base de connaissances baptisés *systèmes experts* [7] [13] [14]. Il s'agit de manipuler des informations factuelles, des règles, des stratégies et du savoir-faire partiel : une approche de résolution de problèmes complexes auxquels il ne peut y avoir de solution algorithmique.

Ces systèmes qui avaient pour premier objectif d'éliminer toute forme de programmation procédurale, n'avaient aucun contrôle, ils étaient caractérisés par l'aspect déclaratif de leur programmation et leur lisibilité. Toutefois, ils ne restent pas à l'abri des difficultés en centralisant dans un système unique l'expression de l'expertise, l'expression de la connaissance et son interprétation.

1.2 Intelligence artificielle distribuée

Dans la vie quotidienne, chaque individu fait appel aux compétences d'experts ou de personnes plus qualifiées pour résoudre un problème d'une certaine complexité. Ces experts peuvent appartenir à un même organisme comme ils peuvent être séparés géographiquement. De cette idée, l'intelligence artificielle est passée du stade classique, où connaissances et expertises sont regroupées dans un seul processus, au stade distribué permettant la coopération de plusieurs experts au sein d'une architecture répartie pour atteindre un objectif commun [2] [6] [1].

L'IA, avec son ouverture sur le monde extérieur, a pu profiter des nouvelles technologies en matière de réseaux et a pu apporter des solutions plus complètes et plus exhaustives aux différents problèmes qui se posent.

1.3 Systèmes multi-agents

L'IAD a conduit à la réalisation de *systèmes multi-agents* [4] [5]. Un agent est une entité autonome, physique (robot) ou abstraite (module logiciel), capable d'interagir avec le monde extérieur ou avec d'autres agents selon une architecture particulière. Dans le premier cas, l'agent dispose d'une représentation partielle de son environnement lui offrant des lois de type social, à savoir la perception, la négociation, la décision, l'action, etc.

Dans le second cas, une entité est plus ou moins "*experte*", elle peut correspondre à une base de connaissances comme elle peut être une procédure spécialisée dans une tâche qui lui est propre. Dans les deux cas, et dans un univers multi-agents, un agent doit être capable de communiquer avec les autres.

Les agents peuvent être soit séparés géographiquement - et dans ce cas le recours à une communication à travers un réseau informatique est nécessaire - soit regroupés selon une architecture particulière basée sur *l'envoi de messages* ou *le partage d'informations*.

1.4 Systèmes distribués d'IA

L'intelligence artificielle recouvre plusieurs disciplines parmi lesquelles nous avons les systèmes experts. Des environnements multi-formalismes ont été conçus pour le développement d'applications à base de connaissances connus sous le nom de *générateurs de systèmes experts*. Ces derniers sont organisés autour d'une architecture intégrée où les composants sont fortement couplés. Avec cette organisation, maintenir une des fonctionnalités du système revient à modifier la totalité de ce dernier, réutiliser un de ses composants s'avère difficile, intégrer un outil de l'extérieur pose non seulement des problèmes de cohérence par rapport à l'ensemble du système mais aussi des problèmes d'occupation d'espace mémoire : ces limites affaiblissent l'efficacité du système. De là, est né le besoin de distribuer les composants d'un GSE sur plusieurs processeurs. Cette architecture a plusieurs caractéristiques :

- *Autonomie* : chaque composant est un processus qui s'exécute de façon indépendante et qui communique avec les autres pour résoudre un problème.

- *Parallélisme* : le système est vu comme un ensemble de processus, où chacun travaille sur sa propre tâche quelle que soit sa localisation.

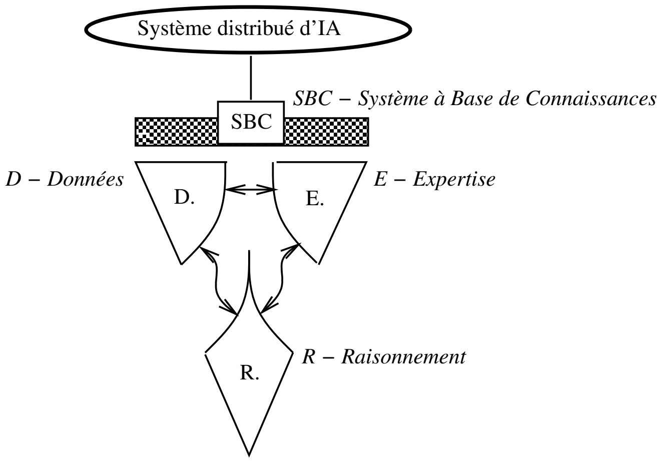
- *Réutilisabilité* : elle permet l'extension progressive du système grâce à sa modularité. Cette extension qui peut être vue comme une évolution des fonctionnalités du système ne se limite pas aux outils d'IA, mais peut couvrir la totalité des disciplines informatiques.

- *Partage des données et des services* : l'intérêt est de permettre à chaque utilisateur quel que soit l'endroit où il se trouve, d'accéder aux données et services de son choix et selon ses besoins.

- *Occupation d'espace mémoire* : les différents composants sont répartis sur des mémoires distinctes. L'espace mémoire qui était réservé pour un système complexe unique, devient le support d'un seul module de ce système.

1.5 Plan de lecture

Dans la première partie, nous présentons brièvement le générateur de systèmes experts sur lequel sont appliqués nos travaux de recherche, à savoir SMECI [9]. Nous décrivons son fonctionnement et nous proposons la manière dont nous



Où pourrions–nous placer un système distribué d'IA ?

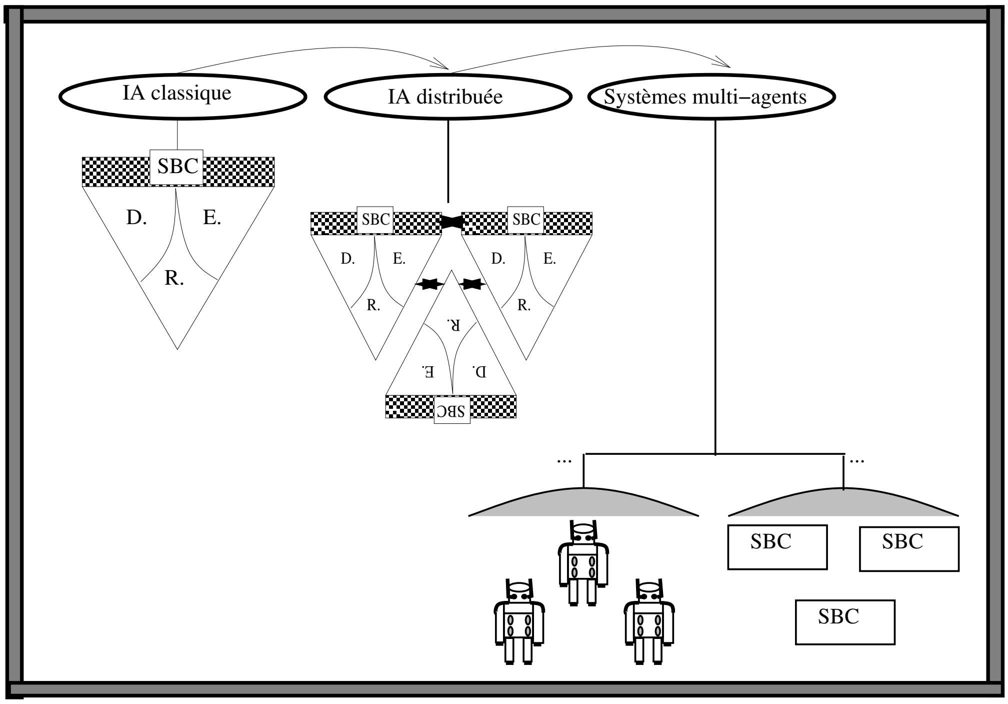


Figure 1 : Evolution de l'IA

séparons ses principales entités. La seconde partie met l'accent sur l'aspect distribué du GSE de point de vue implantation. Les deux dernières parties évoquent le problème de rattachement de l'application en question aux disciplines d'IA, en l'occurrence, l'IAD et les SMA.

2 GSE modulaire : Smeci

2.1 Description de Smeci

Nous considérons un générateur de systèmes experts comme un environnement multi-formalismes de développement d'applications à base de connaissances. Il regroupe les différentes notions de *schémas*, de *règles d'inférence*, de *tâches* et d'*états*.

Le langage de schémas sert à structurer la connaissance et à décrire son comportement, le langage des règles permet de faire des inférences, c'est un moyen d'enrichir l'ensemble des connaissances factuelles. La notion de programmation par tâches offre la possibilité de décomposer un problème complexe en sous-problèmes plus simples et en conséquence, de structurer la base de règles de façon à avoir pour chaque sous-problème l'ensemble des règles qui lui sont associées et qui sont capables de le résoudre.

En cours de raisonnement, Smeci développe un arbre d'états (*mondes multiples*). C'est une trace du raisonnement permettant de mémoriser l'évolution du monde objet. L'arbre d'états permet non seulement de visualiser les chemins qui mènent à des solutions, mais aussi d'expliquer la manière d'y parvenir. La figure 2 présente l'architecture du système.

2.2 Découpage

Tous les composants Smeci sont regroupés dans un seul processus et échangent des informations par envoi de messages. Pour la modularisation du GSE, il est nécessaire de comprendre son fonctionnement. En Smeci, le problème est découpé en sous-problèmes appelés *tâches*. Chaque tâche déclenche une base de règles qui correspond à une liste de règles d'inférence. Le rôle de l'unificateur à ce niveau est d'instancier les variables qui vérifient les prémisses des règles candidates avec la base d'objets. Le résultat de l'unification est une liste de substitutions qui, associées aux actions des règles, engendrent des déductions. Ces déductions modifient la base d'objets et laissent leur trace par construction d'un arbre d'états où chaque état est une liste de transitions enregistrant les déductions élémentaires. L'utilisateur a la possibilité de distinguer les états *solutions* du reste des états.

Nous avons adopté une approche de modularisation qui consiste d'abord, à réaliser un découpage structurel et fonctionnel. Ainsi, nous arrivons à distinguer toutes les structures (agendas, catégories, ...) et fonctions du système

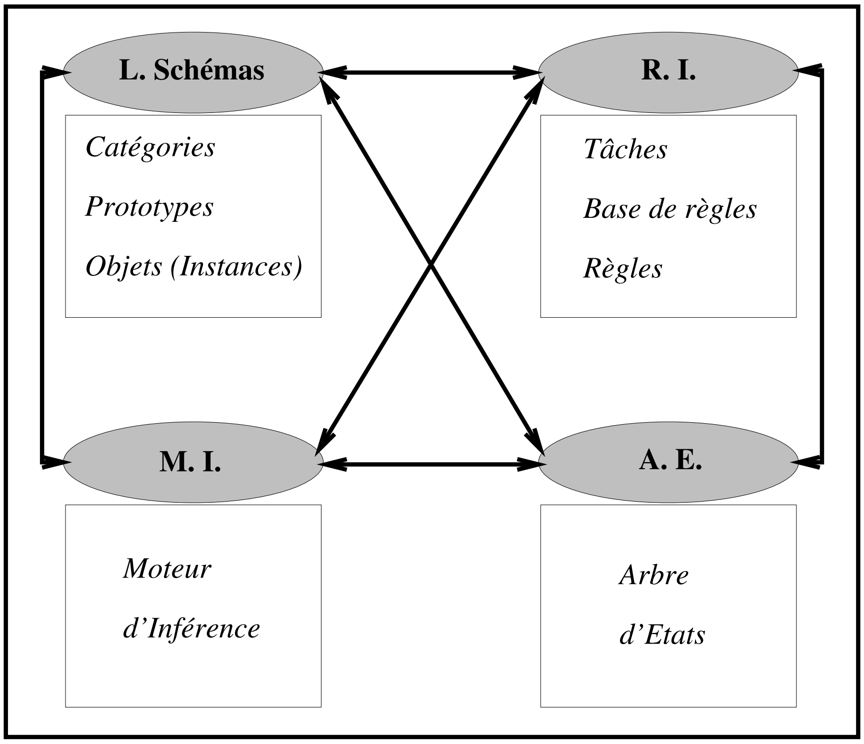


Figure 2 : Principaux composants du GSE

(unification, évaluation, ...). Ensuite, nous définissons des liens de dépendance qui mettent au clair les entrées et les sorties de chaque entité.

A l'issue de cette approche, nous avons pu séparer trois modules principaux présentés sur la figure 3 :

- Module de *représentation des connaissances* qui regroupe toutes les données de l'application. Il joue le rôle d'*un serveur de connaissances* et est constamment à l'écoute aussi bien de l'utilisateur que des modules qui l'entourent.

- Module de *traitement des connaissances*, c'est un composant fondamental de résolution de problème, il est lui-même divisé en deux sous-modules :

  - sous-module d'*expertise*, chargé de modéliser l'ensemble des tâches du problème et les règles d'inférence associées,
  - sous-module d'*unification*, chargé de l'instanciation des prémisses des règles du *module d'expertise*, avec *le serveur des connaissances* pour aboutir à des déductions.

- Module de *trace de traitement*, nécessaire pour chaque application. Il est très important d'avoir une vision claire des différents états construits au cours du raisonnement. Ainsi l'utilisateur a la possibilité de suivre aisément l'évolution de la résolution.

3 Architecture de GSE réparti pour SMECI

3.1 Architecture informatique

Nous proposons une architecture répartie où chaque composant SMECI est un ensemble d'instructions LE-LISP responsable de la résolution d'une tâche bien spécifique. Ces composants sont organisés selon une architecture *client-serveur* et s'exécutent de façon autonome sur des machines distantes [11]. Chaque processus joue le rôle d'un client, d'un serveur ou les deux à la fois.

3.2 Protocole de communication

La répartition des composants du GSE nécessite l'étude d'un protocole d'échange d'informations [17] [12] [15]. Nous avons retenu l'appel de procédure à distance (*RPC*) qui présente un certain nombre d'avantages par rapport aux mécanismes de communication inter-processus :

- Programmation de haut niveau d'abstraction.

- Développement simple.

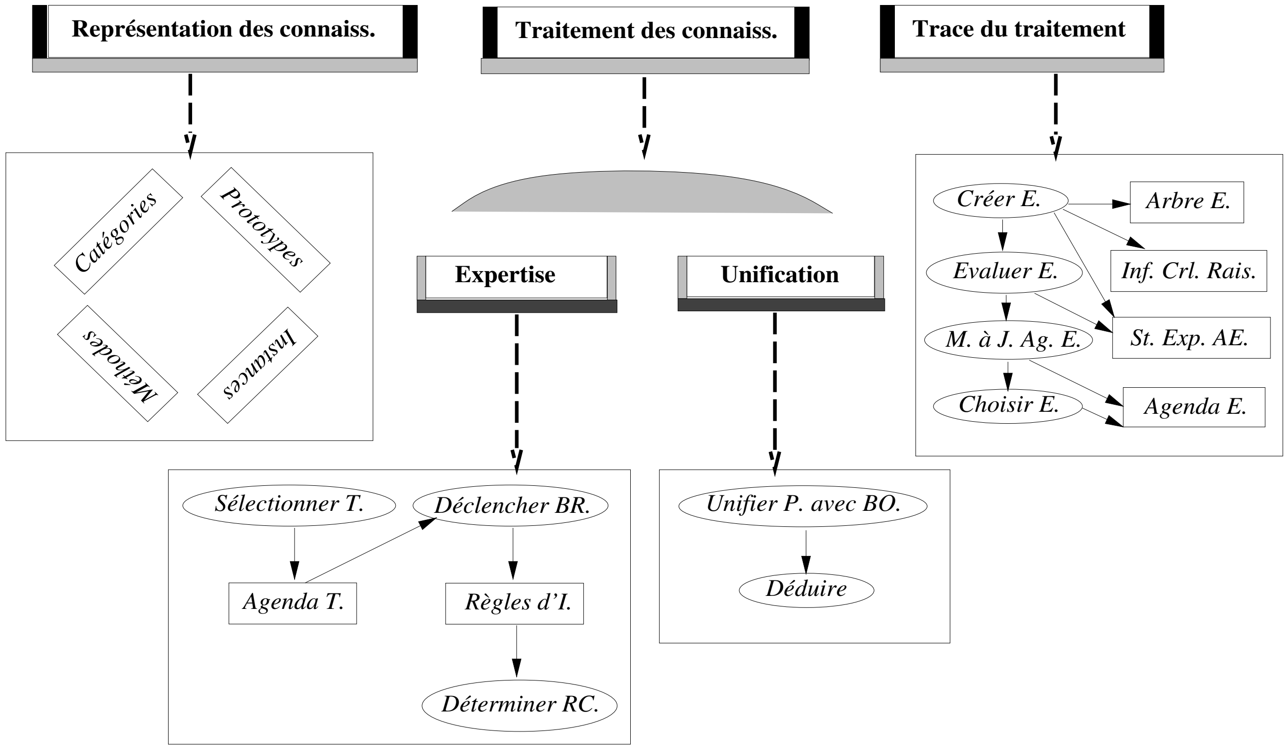


Figure 3 : Découpage de SMECI

- Module bien adapté à l'architecture *client-serveur.*
- Communication synchrone et asynchrone.
- Existence d'une interface fonctionnelle LE-LISP/C [8].

Un autre aspect important du fonctionnement de ce protocole est le fait que le client peut spécifier le temps maximum qu'il désire consacrer à l'exécution d'une requête.

4 GSE réparti comparé à l'IAD

L'IAD s'est toujours concentrée sur la manière de faire coopérer plusieurs expertises pour la résolution d'un problème. Cette résolution nécessite l'intervention de connaissances de plusieurs experts. Ces systèmes aboutissent, grâce au regroupement d'une variété de systèmes à base de connaissances, à des solutions plus sûres, plus exhaustives et plus efficaces.

Les recherches en IA ne se sont pas arrêtées là. Des techniques de *génie logiciel* sont actuellement mises en œuvre pour augmenter la maintenance et la réutilisabilité des systèmes à base de connaissances. Il s'agit de rendre ces derniers modulaires et répartis. *"Les entités expertes"* sont elles-mêmes décomposées en sous-entités réparties. Comment peut-on classer ce genre de système, qui reflète le cas de notre application sur SMECI, par rapport à l'IAD ?

- les exclure de l'IAD paraît brutal, puisqu'ils touchent toujours l'aspect distribué de l'IA,
- les associer à un concept d'*intelligence artificielle distribuée de bas niveau* n'est pas possible, puisqu'un composant n'a pas un comportement intelligent au cours de son traitement,
- les associer à un concept de *système distribué* d'IA semble plus raisonnable. Dans ce cas, nous rattachons l'aspect distribué au système même et non pas à l'intelligence, puisque cette dernière est assez limitée au niveau d'un composant.

5 GSE réparti comparé aux SMA

L'approche IAD consiste à élaborer une architecture constituée d'*entités expertes* indépendantes avec une gestion de conflits éventuels. Les systèmes multi-agents se placent à un niveau plus haut, on ne parle plus d'*entités expertes* mais plutôt de la notion d'*agent* : un terme plus générique qui peut correspondre aussi bien à une base de connaissances, comme il peut être une entité réelle. De là, une définition d'un agent a fait son apparition dans la littérature [5] :

> *Un agent* est une entité réelle (un véhicule, un robot, etc.) ou virtuelle (une entité simulée, un programme, un module, etc.) plongée dans un environnement sur lequel elle est capable d'agir, qui dispose d'une capacité de perception et de représentation partielle de cet environnement (un agent ne peut pas avoir une vue globale sur l'environnement), qui peut communiquer avec d'autres agents et qui possède un comportement autonome conséquence de ses observations, de sa connaissance et des interactions qu'il entretient avec les autres agents.

L'aspect générique d'un agent a conduit les chercheurs à proposer une classification des systèmes multi-agents selon leur comportement et leur degré d'intelligence [4] :

- *Les agents réactifs* sont ceux de plus bas niveau. Ils ne disposent que d'un protocole et d'un langage de communication réduits et leurs capacités répondent uniquement au principe stimulus/action.
- *Les systèmes d'agents communicants*, qui disposent d'un protocole complet de communication. Les agents ne sont pas capables de traiter des informations extérieures à leur domaine d'expertise.
- *Les sociétés de spécialistes*, où chaque agent dispose de compétences précises, de croyances et d'une représentation partielle de son environnement.
- *Les agents intentionnels*, où chaque agent possède des buts explicites et des plans partiels avec la possibilité de contracter des engagements.

Revenons à notre application qui consiste à répartir un générateur de systèmes experts, et voyons ce qu'elle représente par rapport à un système multi-agents. Un composant SMECI est une entité virtuelle : un module qui agit sur l'environnement auquel il appartient, qui communique avec ses voisins et qui possède un comportement autonome. Toutefois, ce module ne dispose d'aucune capacité de perception ni de représentation partielle de son environnement. Cette exception montre qu'une entité SMECI vérifie quelques-uns des concepts caractérisant un agent, et se place – si on se réfère à la classification des SMA – au niveau des *agents réactifs* et *les systèmes d'agents communicants*. Toutefois, cette constatation n'est pas spécifique à SMECI, elle s'applique sur tout *système distribué* qui, par définition, est un ensemble de processus répartis sur plusieurs mémoires et qui communiquent à travers un réseau pour résoudre un problème.

A partir de cette analyse, nous pouvons affirmer qu'un système multi-agents, avec ses lois de type social et le degré d'intelligence relatif à chacun de ses agents, n'est qu'un cas particulier de système distribué. Et donc un système distribué n'est pas forcément un système multi-agents même s'il appartient au domaine de l'IA.

6 Conclusion

Quel que soit le domaine d'intérêt, il est nécessaire et fondamental de bien situer son travail par rapport aux recherches actuelles. Nous nous sommes toujours posés la question suivante : *un générateur de systèmes experts distribué fait-il partie de l'IAD ?* Le but ici est d'enlever l'ambiguïté et d'apporter un élément de réponse en se basant sur les concepts et les définitions auxquels sont parvenus les chercheurs. Il est vrai que le thème abordé est en cours d'exploration mais il est intéressant de regrouper les différents travaux dans les disciplines qui leur correspondent.

Nous pouvons donc conclure que l'absence d'*"intelligence"* et d'*"expertise"* au niveau des composants d'un GSE réparti, ne permet pas de placer ce dernier sous les thèmes de l'IAD et des SMA. Il est plus adéquat de l'associer aux *systèmes distribués* en général et aux *systèmes distribués d'IA* en particulier.

Bibliographie

[1] AAAI. *Eighth National Conference on Artificial Intelligence*, Massachusetts Institute of Technology, Cambridge, 1990. The MIT Press. Volume One. Sessions : Distributed and Parallel Systems, Distributed Artificial Intelligence.

[2] A. H. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann, Palo Alto, CA, 1988.

[3] Cornafion. *Systèmes informatiques répartis: concepts et techniques.* Dunod informatique, Paris-France, 1981.

[4] J. Erceau and J. Ferber. L'intelligence artificielle distribuée. *La Recherche*, 22(233):750 – 758, juin 1991.

[5] J. Ferber and M. Ghallab. Problématiques des univers multi-agents intelligents. In PRC-GRECO, editor, *Actes des journées nationales du PRC-IA*, pages 295–320, Toulouse, mars 1988.

[6] M.P. Gleizes. *Spécification d'une architecture de système multi-expert.* Thèse de doctorat, Université Paul Sabatier de Toulouse (Sciences), octobre 1987.

[7] M. Gondran. *Introduction aux systèmes experts.* Eyrolles, Paris, 1984.

[8] ILOG. *Le-Lisp Communication Inter-Processus (RPC).* Gentilly, France, décembre 1991.

[9] INRIA and Ilog. *SMECI : Le manuel de référence.* Gentilly, 1989.

[10] The Institute of Electrical and Electronics Engineers, Inc. *International Conference on Data Engineering*, California, USA, 1984. IEEE Computer Society Press.

[11] IRISA. *Premières rencontres nationales des jeunes chercheurs en IA*, Rennes, France, septembre 1992.

[12] S.G. Kochan and P.H. Wood. *Unix Networking.* Pipeline Associates Inc, USA, hayden books edition, 1989.

[13] J.L. Laurière. *Intelligence artificielle : résolution de problèmes par l'Homme et la machine.* Eyrolles, Université Pierre et Marie Curie - Paris - France, 1986.

[14] J.L. Laurière. *Intelligence artificielle : représentation des connaissances.* Eyrolles, Université Pierre et Marie Curie - Paris - France, 1988.

[15] SUN Microsystems. *Network Programming Guide.* USA, 1990.

[16] D. Partridge. *Apports de l'intelligence artificielle au génie logiciel.* Manuels Informatiques Masson, Computing Research Laboratory New Mexico State University, 1989.

[17] J.M. Rifflet. *La Communication sous Unix.* McGRAW - HILL, Université Paris VII, 1990.

[18] H. A. Simon. Artificial intelligence: Where has it been, and where is it going. *IEEE Transactions on Knowledge and Data Engineering*, 3(2):128 – 136, june 1991.

[19] A.S. Tanenbaum. *Computer Networks.* Prentice-Hall, Vrije Universiteit Netherlands, second edition, 1988.