

Approches Stratifiées d'Opérationnalisation de Connaissances dans les Systèmes Multi-Agents

Maodo TOURE

Laboratoire API - Université Paul Sabatier

50, chemin des maraîchers

31077 Toulouse - France

Fax : 62.25.88.01

Mots clefs : Systèmes Multi-Agents, Opérationnalisation de Connaissances

Nous proposons une méthodologie de conception de Systèmes Multi-Agents fondée sur l'Opérationnalisation de Connaissances. Ce terme désigne le passage d'un modèle pertinent de connaissances (issu de l'acquisition de connaissances) à un système, qui soit, à la fois :

- 1) persistant par rapport au modèle,*
- 2) et pertinent par rapport à l'application.*

Cet article propose une taxonomie d'agents fondée sur cette notion d'opérationnalisation.

Key Words : Multi-Agent Systems, Knowledge Operationalization

Our work focuses on a methodological framework for building Multi-Agent Systems. The problem is, more precisely, stated in the terms of Knowledge Operationalization ; in other terms, how we can move from a pertinent knowledge acquisition framework to a system which is, both :

- 1) persistent with regard to the knowledge acquisition framework,*
- 2) and pertinent with regard to the problem to be solved.*

The main topic of this paper is the multi-agent taxonomy that this methodology permits to settle.

Introduction

Ce papier soulève la problématique de l'Opérationnalisation de Connaissances dans les Systèmes Multi-Agents (SMAs) ; c'est à dire, comment passer d'un modèle pertinent de connaissances issu de l'Acquisition de Connaissances à un SMA qui soit, à la fois, persistant par rapport au modèle et pertinent par rapport aux objectifs et contraintes de l'utilisateur final.

Cette problématique est encore plus accentuée par le fait que la conception de SMAs se trouve "éclatée" entre deux grandes démarches.

2) **La démarche "formelle" des systèmes d'agents rationnels.** La conception d'agents rationnels fondés sur des états mentaux (ou états Intentionnels) fait appel à des démarches très formelles [Chaib-Draa 90] [Singh 94], très difficiles à conceptualiser et à mettre en oeuvre sur machine [Shoham 93].

1) La démarche "technologique" des premières plateformes multi-agents. Les premières plateformes multi-agents [Gasser & al, 87] [Baujard 92] se sont, le plus souvent, réduites à une certaine forme de technologie de l'Intelligence Artificielle Distribuée et de la robotique. D'abord, elles ne se rattachent à aucune approche méthodologique de conception. Ensuite, l'agent est, pour la plupart du temps, réduit à un simple instrument de parallélisation de tâches de résolution de problème.

Nous proposons une démarche conceptuelle fondée sur l'Ingénierie des Connaissances et, plus précisément, sur l'Acquisition des Connaissances. Elle repose sur les trois composantes principales suivantes.

1) Un support de modélisation de connaissances fondé sur les avancées récentes des approches fonctionnelles d'analyse de connaissances [Chandrasekaran & al, 92].

2) Une organisation multi-agents et les modèles d'agents pertinents par rapport aux rôles de résolution de problèmes et au problème à résoudre [Kornfeld & al, 81].

3) Une architecture d'approches stratifiées d'opérationnalisation de connaissances, permettant de rattacher les différents paradigmes multi-agents [Decker 87] [Ferber & al, 88] à notre démarche méthodologique.

Au plus haut niveau de cette architecture, les agents sont Intentionnels : leurs actes sont conditionnés par leurs états mentaux. Au plus bas niveau, ils sont centrés sur la tâche du domaine : ils sont une forme de parallélisation de la tâche de résolution de problème.

Cette démarche permet, entre autres avantages :

1) de réduire, de façon Verticale, le fossé entre les démarches formelles et les démarches technologiques multi-agents [Ovalle & al, 93],

2) de rattacher, de façon Horizontale, les différents paradigmes multi-agents à une méthodologie complète d'analyse de connaissances [Breuker & al, 94].

Dans ce papier, nous nous intéressons à la troisième composante de la méthodologie ; c'est à dire, l'architecture d'approches stratifiées. Nous nous intéressons, plus précisément, à la mise en oeuvre de l'**organisation multi-agents MAPT¹** selon les différentes approches de l'architecture stratifiée.

MAPT a été proposé dans le cadre de la mise en oeuvre de systèmes intelligents de sécurité². Il insiste sur des contraintes de performabilité³, dans la mise en oeuvre de systèmes multi-agents. Aussi, son support d'analyse de connaissances repose sur un modèle de *tâche* et de *structure de tâche* [Steels 90] fondé sur les avancées récentes de la tâche générique [Chandrasekaran & al, 92] et sur la séparation entre la *tâche* et ses *méthodes de résolution*.

Le papier est structuré en deux grandes sections. La première section présente, de façon détaillée, les différentes composantes de notre démarche : le *modèle de tâche*, l'*organisation multi-agent* et l'*architecture stratifiée*. La deuxième section présente la mise en oeuvre du modèle MAPT selon les différentes approches de l'architecture stratifiée. Quelques remarques relatives aux réalisations effectuées sont, par ailleurs, abordées dans la conclusion.

¹MAPT : Modèle d'Agents Performables pour la résolution de Tâches

²au sens de la protection de l'information

³Performabilité : prise en compte conjointe de la Performance et de la Sûreté

1 Démarche fondée sur l'Opérationnalisation de Connaissances

Nous proposons une démarche simple de conception de Systèmes Multi-Agents. Cette démarche est fondée sur l'opérationnalisation de connaissances ; plus précisément, sur la transformation d'une architecture fonctionnelle de connaissances en une organisation multi-agent. La démarche repose sur les trois grandes étapes suivantes (figure 1) :

- 1) *Acquisition de Connaissances,*
- 2) *Agentification,*
- 3) *Opérationnalisation.*

L'originalité de la démarche réside dans le fait que les agents de résolution de problèmes ne sont, a priori, liés à aucune plateforme d'implantation. Ils correspondent à des agents conceptuels, à l'image des agents *Manager* et *Contractor* du protocole de négociation *Contract Net* [Smith 80] ; ils sont pertinents vis à vis des rôles de résolution de problèmes.

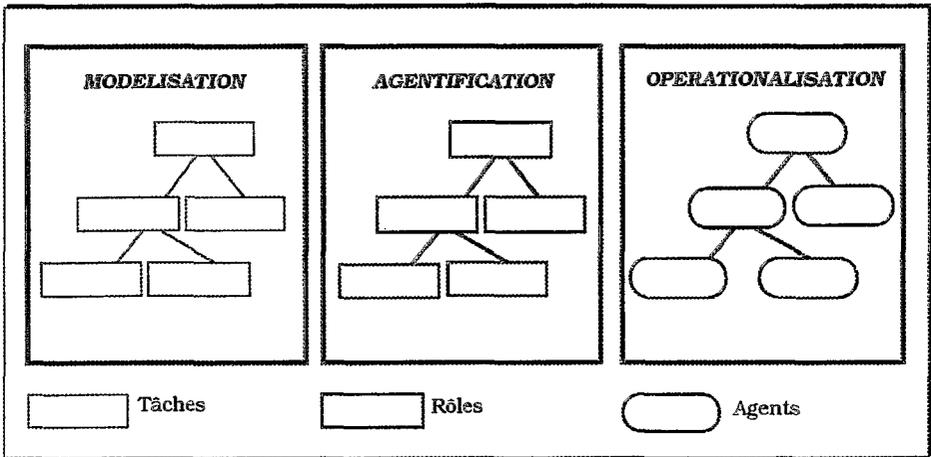


Figure 1. Les trois étapes de la démarche fondée sur l'Opérationnalisation.

Nous rappelons, que le processus de la figure 1 est complété par trois composantes : le *modèle de tâche*, l'*organisation multi-agent* pertinente par rapport au problème à résoudre et l'*architecture stratifiée* d'approches d'opérationnalisation. Ces trois composantes sont présentées, par la suite, avant d'en arriver à celle qui nous intéresse particulièrement ici : l'architecture d'approches stratifiées.

1.1 Le modèle de Tâche

Le modèle de tâche repose sur la séparation entre la tâche et ses méthodes de résolution [Delouis et al, 95] [Chandrasekaran et al, 92]. Cette séparation permet de mettre en évidence, dès la phase de modélisation, la nature stratégique de certaines tâches. Elle est, d'autant plus, souhaitable que le domaine opérationnalisé (la sécurité) fait, essentiellement, ressortir des contraintes de performabilité. Du point de vue de l'Ingénierie des connaissances, la performabilité peut être résumée en une "confrontation opportuniste de sources concurrentes de connaissances".

En guise d'exemple, la figure 2 illustre la modélisation d'une tâche de détection d'intrusions par ce modèle de tâche.

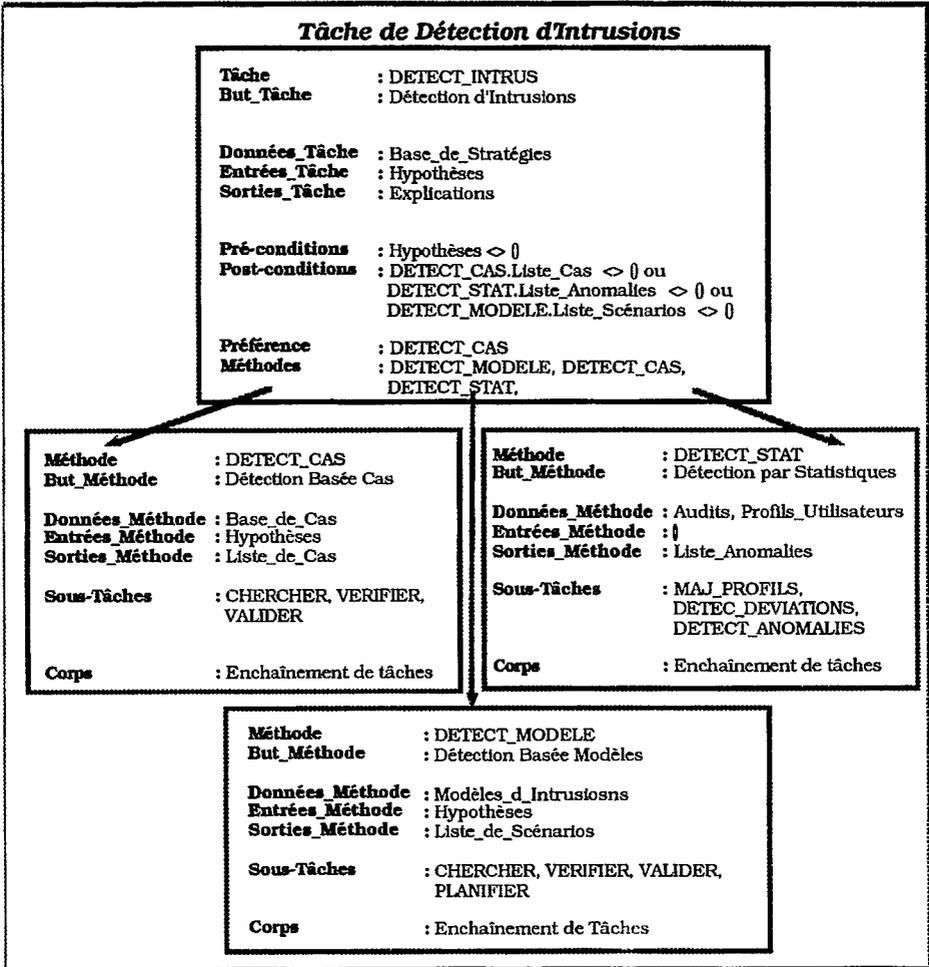


Figure 2. Tâche de Détection d'Intrusions.

La section suivante (§ 1.2) aborde la deuxième composante de la méthodologie ; c'est à dire : l'organisation multi-agent, pertinente par rapport au problème à résoudre. Nous montrerons, plus précisément, comment on peut passer d'une tâche (fondée sur le modèle de la figure 2) à une organisation, a priori, d'agents pertinents par rapport au modèle MAPT.

1.2 De la Tâche à l'Agent : l'organisation MAPT

Le choix de l'architecture organisationnelle MAPT a été guidé par les contraintes du problème à résoudre, qui, sont essentiellement axées sur la performabilité. Les agents MAPT ne correspondent pas nécessairement à des agents réels de résolution de problème, mais plutôt à des rôles que ces derniers peuvent jouer dans la résolution de la tâche, à l'image des agents *Manager* et *Contractor* du protocole *Contract Net* [Smith 80]. MAPT est destiné à des applications dont les exigences en termes de performance et de sûreté sont cruciales, comme la sécurité.

L'architecture organisationnelle d'un système opérationnalisé selon MAPT repose sur trois modèles d'agents (figure 3) : les agents AS, AO, AC.

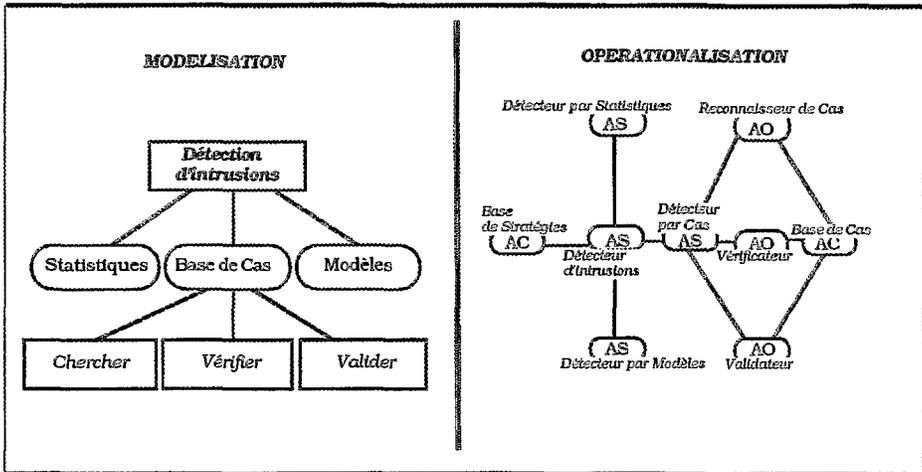


Figure 3. Opérationnalisation de la tâche de *Détection d'Intrusions*.

Ces modèles d'agents correspondent aux différents rôles, pertinents du point de vue de la performabilité, que les acteurs de la résolution du problème peuvent être amenés à jouer dans la résolution d'une tâche :

- 1) L'agent AS assure les tâches stratégiques de confrontation de méthodes de résolution. Il peut être "restreint" à un rôle tactique. L'AS a la charge de "contrôler" les activités des agents AO ou, de façon récursive, des activités d'agents AS.
- 2) L'Agent AC (Agent serveur de Connaissances). L'agent AC est un agent "d'acquisition automatique de connaissances". Son rôle est de pouvoir gérer les connaissances statiques.
- 3) L'agent AO assure les tâches opératoires. Son rôle se limite à l'exécution de tâches élémentaires. Il n'a pas de "pouvoir de décision".

Nous abordons, à la section suivante, la troisième et dernière composante du formalisme : l'architecture d'approches stratifiées. Cette composante permet de rattacher l'organisation MAPT au paradigme multi-agent adéquat.

1.3 Une taxonomie d'agents fondée sur des Approches Stratifiées

On retrouve deux grandes taxonomies d'agents dans la littérature multi-agent : la taxonomie de Decker [Decker 87] et celle de Ferber [Ferber & al, 88].

- 1) La taxonomie de Ferber distingue le paradigme multi-robots du paradigme multi-experts.
- 2) Tandis que celle de Decker repose sur la granularité des "agents", selon que le parallélisme est effectué au niveau machine, au niveau plateforme ou au niveau résolution de problème.

Aucune des deux ne définit, réellement, les agents par rapport à leurs compétences et leurs connaissances.

Nous proposons une taxonomie beaucoup plus orientée vers l'ingénierie des connaissances. Cette taxonomie repose sur une architecture d'approches stratifiées d'opérationnalisation de connaissances. La stratification repose sur le degré d'explicitation du processus de contrôle dans les différents paradigmes multi-agents (figure 4).

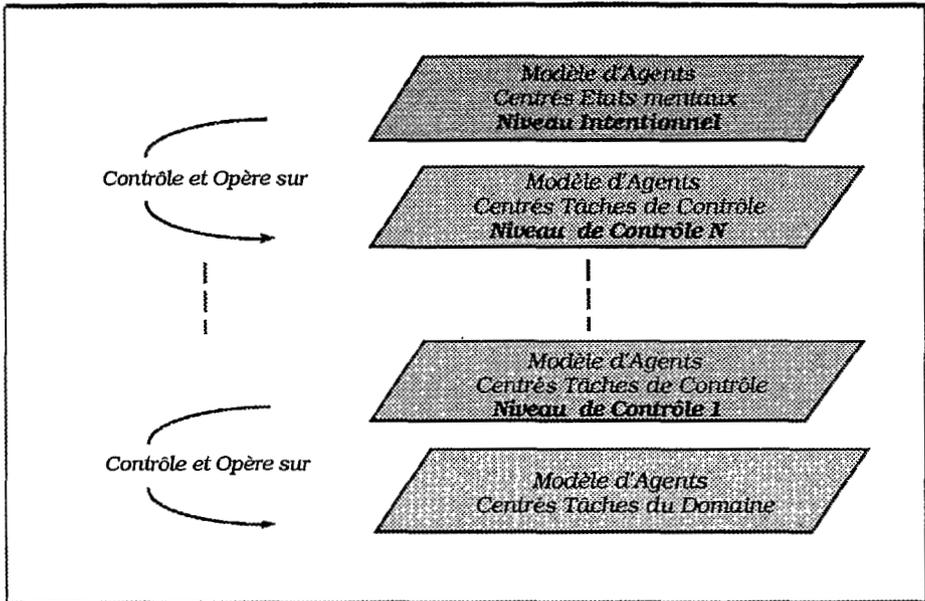


Figure 4. Architecture de Modèles Stratifiés d'Agents Centrés Tâches.

Nous rappelons, ci-dessous, les différents paradigmes de cette taxonomie.

- 1) **Le Niveau Domaine** se situe au plus bas de l'architecture. A ce niveau, les agents sont centrés "Tâches du Domaine". L'implantation y est une forme de "parallélisation" de tâche.
- 2) **Le Niveau Intentionnel** se situe au plus haut niveau de l'architecture. Il représente le plus haut niveau d'explicitation de contrôle. A ce niveau, les agents sont conscients de leurs actes, puis que ces derniers sont conditionnés par leurs états mentaux.
- 3) **Le Niveau MetaⁿContrôle** se situe entre le niveau domaine et le niveau Intentionnel. On peut imaginer plusieurs niveaux imbriqués d'explicitation de contrôle ; cela, en accord avec la définition des métaconnaissances proposée par Pitrat [Pitrat 90]. Pitrat définit les métaconnaissances comme des connaissances permettant de raisonner, de façon récurrente, sur des connaissances ; cette récurrence est notée "métaⁿ" sur la figure 4. L'auteur précise, toutefois, qu'il est difficile d'attribuer une signification aux connaissances, au-delà du niveau "méta¹".

Les différents niveaux de l'architecture sont abordés, en détail et de façon plus formelle, dans la deuxième section de l'article.

2 Approches Stratifiées d'Opérationnalisation de Connaissances

Cette section se propose de montrer, au travers du modèle MAPT, les caractéristiques de chacun des paradigmes d'opérationnalisation.

2.1 Opérationnalisation Centrée Domaine

Comme nous l'avons déjà dit, cette démarche est une certaine forme de parallélisation de la tâche ; la tâche de détection d'intrusions est reprise selon la démarche centrée domaine à la figure 5.

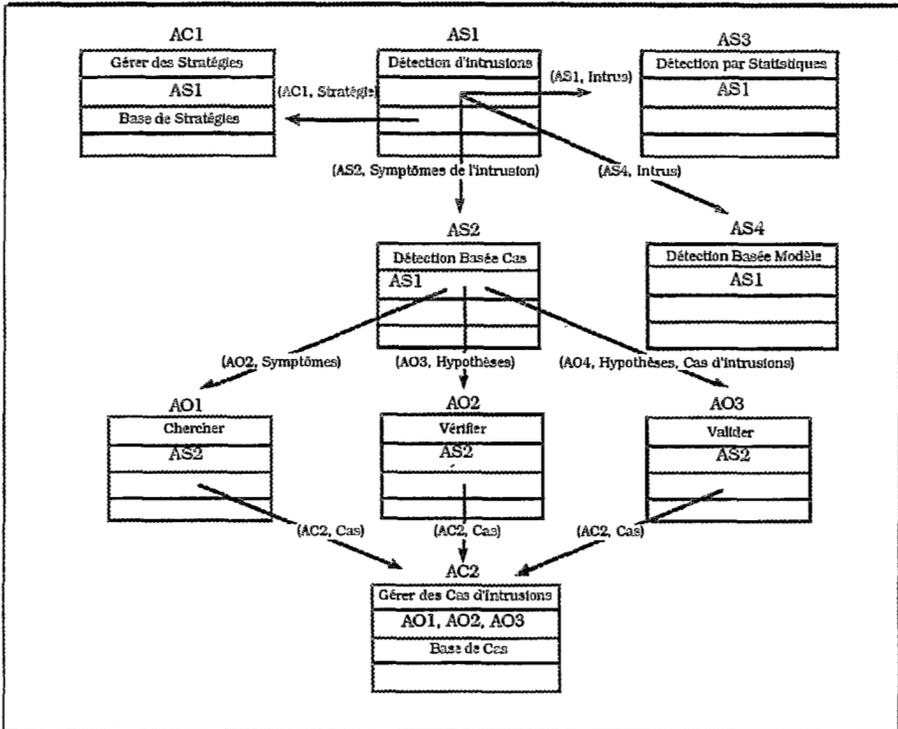


Figure 5. Opérationnalisation Centrée Domaine de la *détection d'intrusions*.

Parmi les propriétés caractéristiques de la démarche centrée domaine, on peut noter que :

- 1) la tâche est "fragmentée" en plusieurs sources de connaissances,
- 2) le processus de contrôle est confondu aux structures de données de contrôle,
- 3) la communication porte sur les connaissances statiques du domaine.

Les propriétés 1 et 2 font que cette démarche ne facilite pas la prise en compte de certaines exigences des Systèmes à Base de Connaissances, telles que : l'explicitation et la coopération. La démarche centrée contrôle tente d'apporter une solution à cette limite.

2.2 Opérationnalisation Centrée Contrôle

Cette démarche repose sur une explicitation du contrôle dans un formalisme, au moins, aussi explicite que celui du domaine. MAPT est illustré, selon cette démarche, à la figure 6.

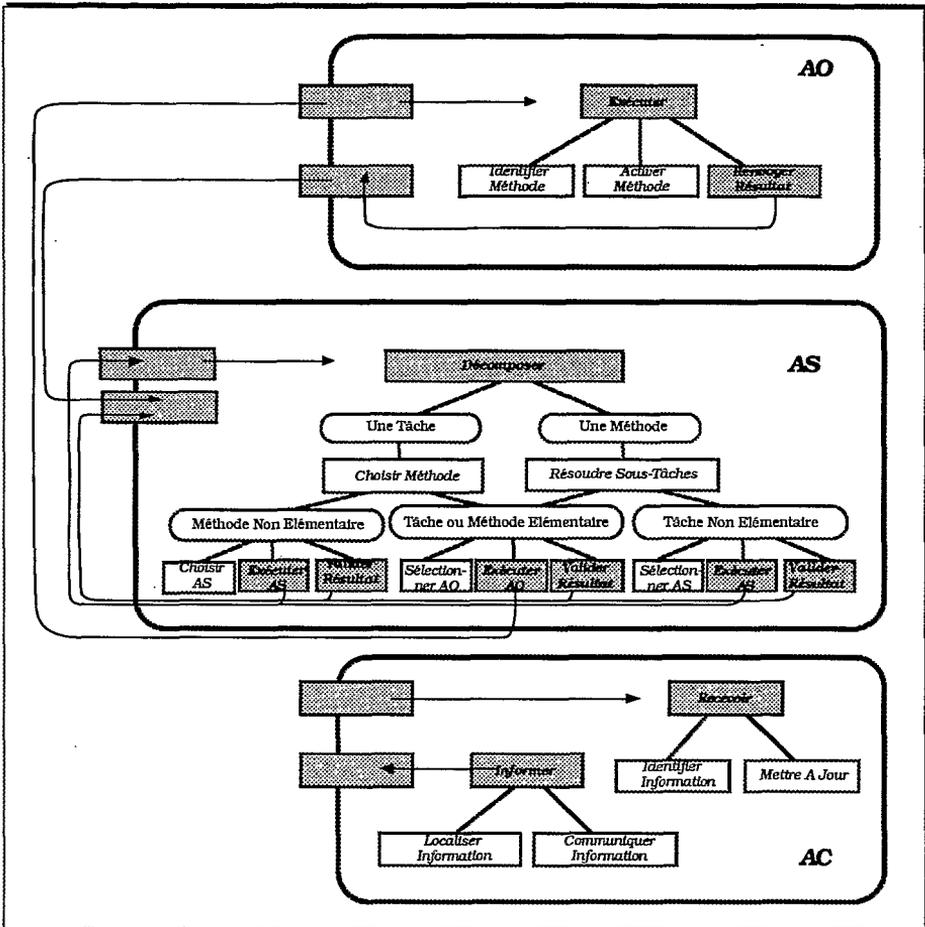


Figure 6. MAPT Centré Contrôle.

Deux propriétés caractérisent cette forme d'opérationnalisation :

- 1) la tâche du domaine est reléguée au plan de structure de données et la communication porte sur toute la structure de tâche,
- 2) les messages de communication sont des tâches (intégrées dans la planification des tâches de contrôle).

De façon récurrente, le processus qui a permis de passer du niveau "domaine" au niveau "meta¹contrôle" peut être généralisé à des niveaux "n-1" et "n" de contrôle.

Cependant, cette démarche souffre de toutes les limites de la planification classique au sens de STRIPS [Fikes & al, 71] ou de NOAH [Sacerdotti 77], en ce sens que le comportement des agents est régi par des plans fonctionnels prédéfinis de tâches.

2.3 Le niveau Intentionnel d'Opérationnalisation

Le formalisme de représentation Intentionnelle que nous allons utiliser repose sur les travaux du philosophe Bratman [Bratman 90] sur la théorie de l'action, en collaboration avec Cohen et Levesque [Cohen & al, 89], visant à considérer l'intention comme attitude mentale "formalisable" à partir des croyances et des buts. L'intention est plus précisément un but persistant permettant, à la limite, de guider le choix des actions (des tâches) de l'agent. Dans le même ordre d'idées, M. Pollack [Pollack 90] définit un formalisme de "plan mental" fondé sur cette notion d'Intentionnalité.

Le plan "mental". Soit Φ , un ensemble d'actes permettant à l'agent i de réaliser un état β . Un "plan mental" est défini de la façon suivante [Pollack 90] [Chaib-Draa 90] :

- (1) i croit que chaque acte de Φ est exécutable, le moment venu,
- (2) i croit que l'exécution des actes de Φ est la meilleure façon d'atteindre le but β ,
- (3) i croit que chaque acte de Φ joue un rôle dans son plan,
- (4) i doit s'engager à atteindre le but β ,
- (5) i doit s'engager à réaliser chaque acte de Φ ,
- (6) i a l'intention de faire chaque acte de Φ qui relève de sa compétence, une fois l'acte qui le précède effectué.

Ces six points sont pris en compte dans le plan mental par le formalisme de la figure 7. Nous avons, plus précisément, repris le formalisme proposé par [Chaib-Draa 90].

PL ($i, \beta, [a_1 \dots a_n]$) \equiv	
Croit($i, Exec(a_k)$) $k=1 \dots n \wedge$	(1)
Croit($i, (Fait(a_1) \wedge \dots \wedge Fait(a_n) \supset \beta)$) \wedge	(2)
Croit($i, Gen(a_k, a_{k+1})$) $\vee \exists e$ Croit($i, Perm(a_k, Gen(e, a_{k+1}))$) $k=1 \dots n \wedge$	(3)
P_But(i, β) \wedge	(4)
P_But($i, Fait(a_k)$) $k=1 \dots n \wedge$	(5)
(Croit($i, Agent(i, a_k)$) \supset IntAct($i, Fait(a_{k-1})?; a_k,$	(6)
P_But($i, Fait(a_{k+1})$)) $k=1 \dots n$	

Figure 7. Le "plan mental" [Pollack 90].

Pour donner une idée de la prise en compte de l'Intentionnalité dans MAPT, nous montrons comment les états mentaux peuvent être directement liés aux primitives tâche et méthode.

Les états Intentionnels liés à la Méthode. Soit M , une méthode permettant à un agent i de réaliser une tâche. Nous proposons le "plan mental" suivant :

- (1) i croit que chaque sous-tâche de M est exécutable, le moment venu,
- (2) i croit que l'exécution des sous-tâches de M est la meilleure façon de la réaliser,
- (3) i croit que chaque sous-tâche de M joue un rôle dans sa résolution,
- (4) i doit s'engager à résoudre la méthode M ,
- (5) i doit s'engager à réaliser chaque sous-tâche de M ,
- (6) i a l'intention de faire chaque sous-tâche de M qui relève de sa compétence, une fois la sous-tâche qui la précède effectuée,
- (7) i croit que pour résoudre une des sous-tâches de la méthode, il doit résoudre le plan de la résolution de la sous-tâche.

```

PLMETHODE (i,M) ≡
  Croit(i,Exec(ST(M,k))) k=1... n ∧ (1)
  Croit(i,(Fait(ST(M,1)) ∧ ... ∧ Fait(ST(M,n)) ⊃ But_Methode(M)) ∧ (2)
  ∃e Croit(i,Perm(ST(M,k),Gen(e,ST(M,k+1)))) k=1... n ∧ (3)
  P_But(i,But_Methode(M)) ∧ (4)
  P_But(i,Fait(ST(M,k))) k=1... n ∧ (5)
  (Croit(i,Agent(i,ST(M,k)) ⊃ IntAct(i,Fait(ST(M,k-1));ST(M,k), (6)
  P_But(i,Fait(ST(M,k+1)))) k=1 ... n ∧
  Croit(i,P_But(i,Fait(ST(M,k)))) ⊃ PLTACHE(i,ST(M,k)) k=1 ... n (7)

```

Figure 8. Le "plan mental" pour la résolution d'une Méthode.

Les états Intentionnels liés à la Tâche. Soit T, une tâche permettant à un agent i de réaliser un but. Nous proposons le plan mental suivant (figure 9) pour résoudre la tâche :

- (1) Au moins une des méthodes de T doit être exécutable, le moment venu,
- (2) i croit que l'exécution d'au moins une des méthodes de T permet de la résoudre,
- (3) i doit s'engager à résoudre la tâche T,
- (4) i doit s'engager à résoudre au moins une méthode de la tâche T,
- (5) i croit que si la méthode j est réalisée, elle a été préférée aux autres,
- (6) i croit que pour résoudre une méthode, il doit résoudre le plan de la méthode.

```

PLTACHE (i,T) ≡
  Croit(i,(Exec(Méthode(T,1)) ∨ ... ∨ Exec(Méthode(T,n)))) ∧ (1)
  Croit(i,(Fait(Méthode(T,1)) ∨ ... ∨ Fait(Méthode(T,n)) ⊃ But_Tâche(T)) ∧ (2)
  P_But(i,But_Tâche(T)) ∧ (3)
  P_But(i,Fait(Méthode(T,1)) ∨ ... ∨ Fait(Méthode(T,n))) ∧ (4)
  Croit(i,Fait(Méthode(T,j)) ⊃ Prefere((Méthode(T,j)) ... (Méthode(T,m)))) ∧ (5)
  Croit(i,P_But(i,Fait(Méthode(T,k)))) ⊃ PLMETHODE(i,Méthode(T,k)) k=1 ... n (6)

```

Figure 9. Le "plan mental" pour la résolution d'une Tâche.

Les figures 8 et 9 représentent les "plans de base" de tâche et de méthode, en ce sens où les agents ne prennent pas en compte la dimension multi-agent dans leur planification. Dans un contexte multi-agent, il devra être complété par la prise en compte de la communication et des relations entre agents ; ces relations ne sont plus fondées sur des rapports Client-Serveur, mais plutôt, sur des rapports de *Sincérité*, de *Serviabilité*.

Conclusion

Le prototypage des plateformes liées aux différents niveaux est actuellement en cours, avec une extension objet d'une version du langage SCHEME⁴ développée au MIT⁵ [Hanson 91]. Cette recherche est par ailleurs appliquée à un domaine concret, la sécurité informatique.

⁴IEEE Standard 1178-1990

⁵Massachusetts Institute of Technology

Références

- [Aussenac & al, 92] N. Aussenac, J.P. Krivine, J. Sallantin. *Editorial du Numéro Spécial Acquisition des connaissances*. In Aussenac, Krivine & Sallantin. Editions RIA, 1992.
- [Baujard 92] O. Baujard. *Conception d'un Environnement de Développement pour la Résol. de Problèmes : Apport de l'IAD et appl. à la vision*. Thèse Univ. J. Fourier, Grenoble 92.
- [Bratman 90] M.E. Bratman. *What Is Intention*. Coh., Morg., Pol., eds. MIT Press, 90.
- [Breuker & al, 94] J. Breuker, W. Van de Velde. *CommonKADS Library for Expertise Modelling. Reusable Problem Solving Components*. IOS Press, 1994.
- [Chaib-Draa 90] B. Chaib Draa. *Contribution à la Résol. Dist. de Problèmes: Une approche basée sur les états Intentionnels*. Thèse de l'Univ. de Valenciennes, Juin 90.
- [Chandrasekaran & al, 92] B. Chandrasekaran, T.R. Johnson, J.W. Smith. *Task-Structure Analysis for Knowledge Modeling*. Com. of ACM, 35 (9), September 1992.
- [Clancey 83] W.J. Clancey. *The Epistemology of Rule-Based Expert System*. AI, 27, 83.
- [Cohen & al, 79] P.R. Cohen, P.R. Perrault. *Elements of a Plan based Theory of Speech Acts*. Cognitive Science, 3(3), 1979.
- [Cohen & al, 89] P.R. Cohen, H.J. Levesque. *Intention is Choice with Commitment*. Artificial Intelligence, 42 (3), 1989.
- [Decker 87] K.S. Decker. *Distributed Problem-Solving Techniques : A Survey*. IEEE Transactions on System Man Cybernetics, SMC-17(5), September 1987.
- [Delouis & al, 95] I. Delouis, J.P. Krivine. *LISA : un langage réflexif pour opérationnaliser les modèles d'expertise*. Revue d'Intelligence Artificielle, 9 (1), 1995.
- [Ferber & al, 88] J. Ferber, M. Ghallab. *Problématiques des univers multi-agents intelligents*. Actes des journées nationales PRC-GRECO Intelligence Artificielle. Toulouse, 1988.
- [Fikes & al, 71] R.J. Fikes, N.J. Nilsson. *STRIPS : A New Approach to the Application of Theorem Proving to Problem Solving*. Artificial Intelligence, 2, 1971.
- [Gasser & al, 87] L. Gasser, C. Braganza, N. Herman. *MACE : A flexible testbed of Distributed AI Research*. Distributed AI, Huhns Ed., Morgan Kauffman, 1987.
- [Hanson 91] C. Hanson. *MIT Scheme Reference Manual*. MIT, January 1991.
- [Kornfeld & al, 81] W.A. Kornfeld, C. Hewitt. *The Scientific Community Metaphor*. IEEE Transactions on System Man Cybernetics, 11, September 1981.
- [Newell 82] A. Newell. *The knowledge level*. Artificial Intelligence, Vol 19(2), 1982.
- [Ovalle 91] A. Ovalle. *Contribution à l'Etude du Raisonnement en Univ. Multi-Agent : KIDS, une Appl. pour l'Interp. d'Images Biomédicales*. Thèse Univ. J. Fourier, Grenoble, 91.
- [Ovalle & al, 93] A. Ovalle, C. Garbay. *Identification Cognit., Modél. et Impl. Techn. : Vers une Méthod. pour la Concept. de SMAs*. Prem. Journ. IAD-SMA. Toulouse, 93.
- [Pitrat 90] J. Pitrat. *Métaconnaissances, Futur de l'IA*. Ed. Hermès 90.
- [Pollack 90] M.E. Pollack. *Plans as Complex mental attitudes*. In Cohen, Morgan & Pollack, editors. The MIT Press, 1990.
- [Sacerdotti 77] E.D. Sacerdotti. *A Struct. for PLans and Behav*. Americ. Else., NY, 77.
- [Schreiber & al, 94] G. Schreiber, B. Wielinga, R. de Hoog, H. Akkermans, W. Van de Velde. *CommonKADS : A Comprehensive Methodology for KBS Development*. IEEE Expert, December 1994.
- [Shoham 93] Y. Shoham. *Agent Oriented Programming*. Artificial Intelligence, 60(1), 1993.
- [Singh 94] M.P. Singh. *Multiagent Systems. Theoretical Framework for Intentions, Know-How, and Communications*. LNAI 799. Springer Verlag, Berlin, 1994.
- [Smith 80] R.G. Smith. *The Contract-Net Protocol : High Level Com. & Control in Distributed Problem Solver*. IEEE Transaction on Computers, C-29(12), 1980.
- [Steels 90] L. Steels. *Components of Expertise*. AI Magazine, 1990.
-