



**TECHNIQUES
DE L'INGÉNIEUR**

Réf. : **H5020 V1**

Systemes multi-agents

Date de publication :
10 décembre 2021

Cet article est issu de : **Technologies de l'information | Technologies logicielles
Architectures des systèmes**

par **Frédéric AMBLARD, Amal
El FALLAH-SEGHRUCHNI, Benoit GAUDOU,
Chihab HANACHI**

Mots-clés

intelligence artificielle |
systèmes complexes |
multi-agents | agent

Résumé Les systèmes multi-agents constituent une discipline issue de l'Intelligence Artificielle Distribuée. Cette discipline offre une approche particulièrement adaptée au traitement de problèmes complexes ayant une nature distribuée. Elle permet l'analyse, la conception et la simulation d'applications distribuées appréhendées comme un ensemble d'entités relativement autonomes (agents), capables de raisonner, de s'organiser, d'interagir et de s'adapter à leur environnement. L'objectif de cet article est de fournir une vue synthétique de cette discipline. Il présente le contexte historique dans lequel elle est apparue, les fondements et les définitions associés et ses domaines d'applications actuels. Il explique également le fonctionnement interne des agents, leurs raisonnements et leurs propriétés. [...]

Keywords

artificial intelligence | complex
systems | multi-agents | agent

Abstract Multi-agent systems is a discipline that grew out of Distributed Artificial Intelligence. This discipline offers an approach particularly suited to cope with complex problems having a distributed nature. It is appropriate for the analysis, design and simulation of distributed applications understood as a set of relatively autonomous entities (agents), able to reason, organize, interact and adapt to their environment. The objective of this article is to provide a synthetic view of this discipline. It presents the historical context in which it appeared, the foundations and associated definitions and its current fields of application. It also explains the internal behavior of agents, their reasoning and their properties. It develops their modes of interaction and organization and exposes their capacity for learning. [...]

Pour toute question :

Service Relation clientèle
Techniques de l'Ingénieur
Immeuble Pleyad 1
39, boulevard Ornano
93288 Saint-Denis Cedex

Par mail :

infos.clients@teching.com

Par téléphone :

00 33 (0)1 53 35 20 20

Document téléchargé le : **27/09/2022**

Pour le compte : **7200083408 - universite grenoble alpes // 130.190.247.198**

© Techniques de l'Ingénieur | tous droits réservés

Systemes multi-agents

par **Frédéric AMBLARD**

*Professeur, Université Toulouse 1 Capitole,
Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505, Toulouse, France*

Amal EI FALLAH-SEGHROUCHNI

*Professeure, Sorbonne Université,
Laboratoire d'Informatique de Paris 6, UMR CNRS 7606, Paris, France
Directrice du Centre international d'intelligence artificielle du Maroc, Ai movement*

Benoît GAUDOU

*Maître de conférences, Université Toulouse 1 Capitole,
Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505, Toulouse, France*

et **Chihab HANACHI**

*Professeur, Université Toulouse 1 Capitole,
Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505, Toulouse, France*

1. De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents.....	H 5 020 - 3
2. Définition, propriétés et applications des agents	— 3
2.1 Qu'est-ce qu'un agent ? Un système multi-agents ?	— 3
2.2 Propriétés des systèmes multi-agents	— 4
2.3 Grandes classes d'application des SMA	— 4
3. Architectures, fonctionnement et raisonnement d'un agent.....	— 5
3.1 Typologie et architectures des agents	— 5
3.2 Agents BDI, figure de proue des SMA cognitifs.....	— 5
4. Interactions entre agents	— 7
4.1 Interactions et langage de communication	— 7
4.2 Protocoles d'interaction.....	— 7
4.3 Coordination.....	— 7
5. Organisation multi-agents	— 9
6. Apprentissage en univers multi-agents	— 10
7. Ingénierie des SMA : méthodes et outils	— 11
7.1 Méthodologies	— 11
7.2 Outils	— 11
8. Conclusion.....	— 12
9. Sigles, notations et symboles.....	— 12
Pour en savoir plus	Doc. H 5 020

Les Systèmes Multi-Agents (SMA) sont utilisés pour concevoir, modéliser, analyser et simuler des systèmes complexes impliquant de multiples entités en interaction, dont la coordination permet d'atteindre un objectif collectif ou de favoriser la stabilité du système. Ces entités autonomes composant un SMA, appelées agents, peuvent être distribuées et hétérogènes, et agir dans un environnement ouvert et évolutif. La discipline des SMA apporte les concepts, théories et outils nécessaires pour appréhender de tels systèmes avec le bon niveau d'abstraction.

Ces vingt dernières années, cette discipline a connu des avancées remarquables sous l'effet conjugué de plusieurs facteurs. En premier lieu, il faut noter l'ouverture de cette discipline à d'autres, en particulier aux sciences humaines et sociales (géographie, sociologie, linguistique, psychologie cognitive, économie et plus récemment droit), à la biologie et aux mathématiques. Motivée au départ par des applications, cette ouverture permet aujourd'hui aux SMA de constituer un domaine de recherche pluridisciplinaire s'appuyant sur des connaissances théoriques solides, validées et partagées par une communauté. Ce domaine a aussi su continuellement exploiter les résultats de l'Intelligence Artificielle (logique, apprentissage artificiel) dont lui-même est issu. Parallèlement, des progrès spécifiques ont été réalisés en termes d'ingénierie par la proposition de méthodes (GAIA, MOISE...), de formalismes basés sur la logique, de notations (ex. AUML), de langages standards (ex. FIPA-ACL) mais aussi de plateformes de simulation (GAMA, NetLogo) ou de développement (Madkit, JADE) permettant une industrialisation des applications. Cette évolution a aussi été rendue possible grâce à une communauté scientifique particulièrement dynamique, organisée autour de projets fédérateurs dont le réseau d'excellence européen AgentLink, les conférences internationales annuelles AAMAS et IJCAI et les revues JAAMAS, IJAOS et JASSS.

Cette maturité permet aujourd'hui de concevoir des agents capables de raisonner, coopérer, s'organiser, agir, anticiper, apprendre et s'adapter aux changements de leur environnement, possiblement évolutif. Ces capacités dotent les SMA d'un pouvoir d'expression riche, aussi bien au niveau social que cognitif, permettant d'aborder des problèmes complexes dans des domaines aussi divers que les systèmes ambiants intelligents, la robotique collaborative, les chaînes logistiques ou la simulation sociale... À titre d'exemple, il peut s'agir d'une flottille de drones se coordonnant pour surveiller l'évolution d'une catastrophe naturelle et assistant les acteurs sur le terrain et la cellule de crise, d'une équipe de robots footballeurs capable d'en affronter une autre (cf. compétition robocup) en mettant en œuvre une stratégie. Cela peut également être des entités logicielles simulant les acteurs d'un réseau social et leurs émotions et conduisant à la découverte et compréhension de phénomènes de groupe émergents et non anticipés (par exemple la panique). Ses domaines d'application se sont élargis du fait du changement physique de notre environnement avec l'Internet des Objets sur des espaces de vie communs (Parc, Campus, Ville), les megadonnées produites par les réseaux sociaux ou des phénomènes planétaires (catastrophes naturelles, épidémies) suffisamment complexes pour exclure toute approche naïve et centralisée afin de les comprendre, les analyser ou les simuler.

La suite de cet article est organisée comme suit. La première section expose le contexte historique dans lequel sont apparus les SMA. La section 2 définit les notions d'agent et de SMA, discute les propriétés des agents et détaille les grandes classes d'application des SMA. La section 3 explique le fonctionnement interne des agents, leur architecture et leur raisonnement. Les sections 4 et 5 détaillent deux dimensions sociales des SMA, qui en font l'originalité : leurs modes d'interaction, et d'organisation possiblement émergente. La section 6 expose un thème particulièrement dynamique actuellement : l'apprentissage en univers multi-agents. La section 7, consacrée à l'ingénierie des SMA, présente des méthodes de conception, des langages standards et des plateformes de simulation et de développement.

1. De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents

L'Intelligence Artificielle (IA) a été confrontée à des difficultés pour aborder et résoudre efficacement des problèmes qui exigent la mise en commun et l'échange de plusieurs points de vues ou impliquent des connaissances, des traitements ou un contrôle distribués. L'Intelligence Artificielle Distribuée (IAD) est née dans les années 1970 pour simplifier et rendre plus efficace la résolution de tels problèmes par une conception modulaire des solutions et leur exécution distribuée et coopérative. Un des premiers résultats fondateurs, significatif de ce changement de paradigme, est le développement du système de reconnaissance de la parole Hearsay II [24] qui exige la combinaison de connaissances spécialisées (analyse du signal, acoustique, syntaxe, sémantique et pragmatique). Il est conçu autour d'une architecture de type tableau noir. Dans ce modèle, les connaissances sont organisées en plusieurs modules indépendants, en fonction de leur spécialité. Ces modules appelés **sources de connaissances** font avancer la résolution d'un problème en échangeant des informations (hypothèses, problèmes, solutions) de manière opportuniste via un espace d'échanges partagé : le tableau noir. Un dispositif de contrôle centralisé gère les possibles conflits d'accès au tableau noir. Ce modèle a évolué et a été exploité par la suite pour divers autres types de problèmes [21].

L'IAD se déclinait initialement en trois axes : l'IA parallèle, la résolution coopérative de problèmes et les systèmes multi-agents. On trouvera une discussion très approfondie des différences entre les deux derniers axes dans [46].

L'IA parallèle s'est intéressée aux problèmes pouvant se décomposer en sous-problèmes indépendants et elle est donc adaptée à un traitement parallèle sur des machines multiprocesseurs. Le résultat final est une simple composition des résultats partiels. La motivation principale du passage à l'IAD ici est le gain de performance. On est très proche de systèmes distribués conventionnels, avec des nœuds qui sont des processeurs capables de résoudre des problèmes par une approche IA.

La résolution coopérative de problèmes (DPS en anglais, pour *Distributed Problem Solving*) s'est intéressée à des problèmes décomposables en sous-problèmes interdépendants confiés à des agents coopératifs et conçus avec un même but global. Les agents n'ont pas la liberté de refuser de coopérer. La motivation ici est d'avoir un système robuste grâce à des techniques prédéfinies de coordination de tâches et d'échanges d'information, permettant la convergence efficace du système vers l'objectif global.

Les **systèmes multi-agents** [25], quant à eux, mettent en jeu des agents autonomes (possiblement réalisés par des concepteurs différents), ayant des objectifs individuels et capables de coopérer ou d'entrer en conflit pour les atteindre. Ces agents peuvent préexister indépendamment les uns des autres avant d'entrer en jeu dans un même SMA afin de participer à un objectif collectif. L'inspiration vient ici des organisations humaines et l'objectif est celui d'obtenir des sociétés d'agents capables d'interagir et s'organiser avec des mécanismes de haut niveau et de raisonner de manière rationnelle. Cela suppose qu'ils aient une représentation de l'environnement, des autres agents et d'eux-mêmes. Cependant, l'autonomie octroyée aux agents peut engendrer du désordre et un dysfonctionnement global ne permettant donc pas d'atteindre l'objectif assigné au système. Ce constat a conduit à introduire des modes de régulation et de coordination divers à travers des concepts et techniques, inspirés de la sociologie ou de la théorie des organisations, tels que les conventions, normes, contrats, engagements, négociation, protocoles d'interaction, structures organisationnelles... Ce sont autant de nouvelles problématiques de recherche qui se sont ouvertes et ont renouvelé le domaine de l'IA.

Parallèlement à cette vision sociologique des agents, une vision bio-inspirée s'est affirmée pour défendre l'idée qu'à partir d'une multitude d'agents au comportement simple pouvaient émerger des formes d'intelligence collective (organisations, formes d'interactions) à l'image d'une ruche d'abeilles ou d'une fourmilière. Il existe naturellement tout un continuum de structures d'agents possibles entre ces deux extrêmes et ces structures peuvent aussi se combiner. On trouvera dans [21] une discussion approfondie sur ces différentes visions des SMA.

Sur cette base scientifique commune, les SMA se sont développés sur plusieurs fronts qui seront présentés dans la suite : élargissement des domaines d'application, intégration de travaux issus d'autres disciplines, extension des propriétés des agents, ingénierie logicielle... Sur ce dernier front, il faut mentionner que les SMA constituent également aujourd'hui un paradigme de programmation à part entière : on parle de **Programmation Orientée Agents**.

À retenir

- **L'Intelligence Artificielle Distribuée (IAD)** est apparue dans les années 1970 pour rendre plus efficace la résolution de problèmes, impliquant des connaissances, des traitements ou un contrôle distribués, par une conception modulaire des solutions et leur exécution distribuée et coopérative.
- Les **systèmes multi-agents**, branche de l'IAD, mettent en jeu des agents autonomes, ayant des objectifs individuels et capables de coopérer ou d'entrer en conflit pour les atteindre.

2. Définition, propriétés et applications des agents

2.1 Qu'est-ce qu'un agent ? Un système multi-agents ?

2.1.1 Agent

Défini prosaïquement, **un agent est une entité agissante** [15] [57]. Ce qui peut passer pour une tautologie permet conceptuellement de distinguer un agent d'un objet issu des approches orientées objet. Là où un objet possède des comportements qui sont exécutés suite à des envois de message (appel de méthodes), un agent possède, au moins conceptuellement, une autonomie et une pro-activité qui font que celui-ci est à l'origine de ses propres actions [38].

Sur quoi agit-il ? Un agent agit sur l'environnement qu'il partage avec d'autres agents. Cet environnement est compris comme l'ensemble des entités (objets ou agents) extérieures à l'agent et que celui-ci peut percevoir (via ses senseurs) et dont il peut changer l'état (via ses effecteurs) ou avec lesquels il peut communiquer (dans le cas des autres agents).

Ainsi, l'activité d'un agent est fréquemment organisée en trois phases : perception, cognition, action. La phase de perception permet à l'agent de récupérer l'ensemble de l'information mise à jour concernant son environnement. La phase de cognition permet d'une part de mettre à jour, en prenant en compte ses perceptions, la représentation qu'il se fait de son environnement et fait évoluer l'état de l'agent en conséquence. La phase de cognition inclut d'autre part une forme de prise de décision qui permet à l'agent de déterminer la ou les actions qu'il doit réaliser.

Cette phase de cognition peut prendre, en fonction des contextes et des applications, des formes très différentes qui peuvent aller de formes réduites de cognition (approches réactives) jusqu'à des

formes plus élaborées (approches cognitives) incluant par exemple des croyances, des objectifs, des plans [54]... C'est également cette phase qui conduit souvent à qualifier les agents d'entités intelligentes [70], ce qui n'est sans doute qu'une propriété très secondaire des approches multi-agents, la principale ne résidant pas réellement dans l'intelligence des constituants de ces systèmes mais dans l'intelligence collective observée à l'échelle du système.

Pour synthétiser, Ferber [25] appelle **agent** « une entité physique ou virtuelle (a) qui est capable d'agir dans un environnement, (b) qui peut communiquer directement avec d'autres agents, (c) qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser), (d) qui possède des ressources propres, (e) qui est capable de percevoir (mais de manière limitée) son environnement, (f) qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune), (g) qui possède des compétences et offre des services, (h) qui peut éventuellement se reproduire, (i) dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit. »

2.1.2 Système multi-agents

Un **système multi-agents** est par définition un système composé de plusieurs agents en interaction, au sein d'un environnement.

Les agents d'un SMA font système au sens où ils forment un ensemble cohérent autour d'un objet commun (problème à résoudre, environnement partagé...). Le comportement du système résulte des interactions entre les comportements individuels des agents. On peut distinguer deux grands types d'approches multi-agents. Il y a d'une part les systèmes dirigés par un but collectif, typiquement un problème à résoudre. Dans ce cas, chaque agent pris isolément n'a pas les capacités de résoudre le problème, il n'y a pas de système de contrôle global centralisé, les données sont distribuées et l'exécution est asynchrone [38]. On retrouve par exemple dans cette catégorie la résolution de problèmes distribués (coordination d'une flotte de satellites) ou la robotique collective (les robots footballeurs). D'autre part, dans une approche plus descriptive, les sociétés artificielles, souvent simulées, pour lesquelles on cherche davantage à comprendre l'articulation entre comportements individuels et régularités collectives observées. On retrouve en particulier dans cette catégorie la simulation sociale multi-agents (par exemple simulation de la réaction d'une population face à un aléa, incendie ou inondation). Pour ces dernières, l'ensemble des agents n'a typiquement pas un objectif à remplir collectivement.

Il est à noter que cette relation agent-système multi-agents peut être employée à plusieurs échelles, un système multi-agents pouvant être lui-même encapsulé dans l'agent d'un système multi-agents de plus haut niveau. On débouche ainsi sur les **approches SoS** (*System of Systems*) [44].

2.2 Propriétés des systèmes multi-agents

Par construction, les systèmes multi-agents sont particulièrement bien adaptés aux problèmes distribués. Parmi leurs capacités sociales, les agents peuvent faire évoluer leur organisation (rompre ou créer des relations avec d'autres agents ou en choisissant un paradigme un peu différent, créer des groupes ou les quitter). En conséquence, on qualifie souvent les systèmes multi-agents d'**auto-organisés**, l'organisation du système étant modifiée par les agents eux-mêmes et évoluant en fonction de l'évolution

de l'environnement ou du problème à résoudre. L'ajout (ou le retrait) de nouveaux agents peut ainsi être réalisé de manière flexible, permettant aux systèmes multi-agents d'être utilisés dans des **systèmes ouverts** (l'ajout et le retrait de satellites dans le cadre de la coordination d'une flotte impliquent uniquement une réorganisation du système multi-agents). L'**auto-adaptation** continue des agents et du système multi-agents à son environnement concourt également à cette adéquation avec les systèmes ouverts.

Enfin, la propriété principale des systèmes multi-agents tient à l'articulation entre comportements individuels et régularités ou objectifs collectifs. Par construction, le concepteur d'un système multi-agents définit son système à l'échelle individuelle (microscopique) pour résoudre des problèmes ou reproduire des régularités à l'échelle collective (macroscopique). Les régularités macroscopiques observées sont produites par **émergence** à partir des comportements individuels et de leurs interactions. Le concepteur du système ne maîtrise pas complètement cette émergence, et les approches de validation ou de démonstration formelles des propriétés macroscopiques ne sont jusqu'à présent que partielles et limitées malgré des résultats pratiques convaincants. Pour synthétiser la situation de manière caricaturale : en pratique ça marche, mais en théorie on ne parvient pas toujours à le démontrer.

2.3 Grandes classes d'application des SMA

Pour présenter de manière synthétique les domaines d'applications des systèmes multi-agents, nous aurons recours à 6 grandes classes d'application :

- la **simulation de systèmes complexes** : en particulier de systèmes socio-environnementaux, pour lesquels une approche de modélisation à base d'agents présente l'avantage de pouvoir raisonner sur les comportements individuels, qu'il s'agisse de décrire le fonctionnement du système ou de construire des solutions concernant la gestion de ce système. Dans ce domaine, les plateformes de simulation, en particulier GAMA ou Netlogo, permettent à l'utilisateur/modélisateur de réduire l'effort d'implémentation à la seule partie du modèle ;

- l'étude théorique de **dilemmes sociaux** : souvent capturés par des approches formelles proches de la théorie des jeux ou du choix social, il s'agit bien souvent de modéliser formellement ces dilemmes et d'en étudier les propriétés de manière analytique ;

- la **modélisation à base d'agent** : un effort important de la recherche sur les systèmes multi-agents consiste à enrichir et renforcer les langages de modélisation, par exemple par le développement de DSL (*Domain specific Language*), la construction d'ontologies ou le travail spécifique sur des concepts centraux des multi-agents tels que la coordination ou la coopération ;

- la **résolution collective de problèmes** : concernant les problèmes par nature distribués, l'effort des systèmes multi-agents rejoint d'autres approches en recherche opérationnelle comme les algorithmes fournis ou l'optimisation par essaim particulaire (PSO, pour *Particle Swarm Optimization*) ;

- les **agents logiciels** : il s'agit dans ce cadre de proposer des architectures et des solutions logicielles nécessaires pour le déploiement effectif de systèmes multi-agents conceptuels et leur distribution physique (sur un réseau par exemple), des langages d'acteurs aux agents mobiles, en passant par les systèmes embarqués ;

- l'**humain dans la boucle** : correspond aux approches centrées sur le profilage utilisateur, les assistants et les agents conversationnels, permettant d'aller au-delà de l'élicitation et la prise en compte des préférences pour permettre également la coordination entre agents assistants. Un exemple typique est la coordination entre agents assistants pour permettre la coordination d'agents humains et réserver un restaurant un soir de la semaine (prise en compte des agendas, des préférences et contre-indications de chacun, mise en

place du plan : réserver le restaurant, trouver une baby-sitter pour la soirée...).

À retenir

- Un **agent** est une entité « située dans un environnement et capable d'actions autonomes dans cet environnement afin d'atteindre ses objectifs. » [69].
- Un **système multi-agents** est par définition un système composé de plusieurs agents en interaction au sein d'un environnement partagé et dynamique.
- Les agents d'un SMA sont capables de communiquer (ou plus généralement d'interagir) pour s'(auto-)organiser et faire émerger des propriétés (par exemple résoudre un problème ou reproduire le comportement d'un système complexe).

3. Architectures, fonctionnement et raisonnement d'un agent

Au sein d'un SMA, les agents sont très souvent hétérogènes : ils peuvent être de natures différentes (logiciels ou physiques, par exemple des robots), être coopératifs ou compétitifs, ou avoir des capacités de raisonnement et des architectures différentes (réactifs vs cognitifs).

3.1 Typologie et architectures des agents

Les différentes capacités de raisonnement (i.e. de délibération) des agents induisent une typologie et une architecture associées. Traditionnellement, on distingue deux archétypes principaux d'agents : les *agents réactifs* et les *agents cognitifs*. Néanmoins, la plupart des agents existants se situent plutôt sur un continuum entre ces deux archétypes.

Les **agents réactifs** ont un comportement réflexe, lié à leurs perceptions, de type stimulus/action.

Un SMA constitué d'agents réactifs possède généralement un grand nombre d'agents (des milliers, voire des millions d'agents) et le pari, dans ce type de SMA, porte sur l'émergence d'un comportement collectif intelligent, à partir de cet ensemble d'agents dotés d'une intelligence individuelle limitée. La plupart des recherches dans le domaine des agents réactifs s'inspirent des travaux de Brooks [15]. Le cycle de contrôle de tels agents est illustré par la figure 1.

À l'opposé, les **agents cognitifs** possèdent une représentation partielle mais sophistiquée de leur environnement, ont des buts explicites et sont capables de planifier leur comportement, de mémoriser leurs actions passées, de communiquer par envoi de messages ou via des langages d'interaction élaborés, de négocier, etc.

Un SMA constitué d'agents cognitifs possède généralement un nombre restreint d'agents (des dizaines ou des centaines d'agents), mais chacun est capable d'effectuer des tâches de haut niveau. Le cycle de contrôle de ces agents cognitifs est illustré sur le cas des agents cognitifs BDI (cf. § 3.2.2).

```
/*cycle de contrôle d'un agent réactif*/
Variables
rules : règles de type condition-action
percepts : ensemble de perceptions
Traitement
Répéter {
  percepts := observer(); /*perception de l'environnement*/
  state := interpréter_perceptions(percepts); /*mise à jour
  de son état interne*/
  rule := choix_règle(state,rules); /*choix d'une règle exécutable*/
  exécuter(rule[action]); /*exécution de la partie action de la règle*/
} Indéfiniment ;
```

Figure 1 – Cycle de contrôle d'un agent réactif

Les agents cognitifs peuvent être de natures différentes selon leur mode de fonctionnement. Ils peuvent être *intentionnels*, i.e. dotés d'attitudes intentionnelles telles que les croyances, les désirs et les intentions ; *rationnels*, i.e. agissant selon une rationalité donnée comme la rationalité économique ; *normés*, i.e. agents évoluant dans un système doté de normes sociales.

3.2 Agents BDI, figure de proue des SMA cognitifs

La cognition fait référence aux aptitudes cognitives d'un agent telles que l'apprentissage, le raisonnement, la prise de décision, la planification autonome, etc. Ces capacités permettent aux agents de s'adapter aux changements potentiels de leur environnement. Ces aptitudes sont à considérer dans un contexte collectif multi-agents (par exemple, apprentissage multi-agents, décision distribuée, connaissances communes et/ou distribuées, etc.) car c'est bien le passage du niveau individuel au niveau collectif qui caractérise les SMA et ouvre de larges champs de recherche, y compris à d'autres communautés d'IA.

Un SMA cognitif est particulièrement adapté à la mise en œuvre de systèmes complexes du fait de :

- la décomposition et la répartition des connaissances et des mécanismes de traitement grâce au concept d'agent ;
- la dynamique du contrôle de résolution d'un problème distribué grâce à l'organisation dynamique et l'affectation des tâches aux agents modifiables en cours de résolution ;
- l'aptitude des agents, au sein d'un SMA, à traiter des problèmes simultanés et potentiellement corrélés avec des possibilités d'optimisation ;
- l'adaptation et la possibilité d'apprentissage des agents qui leur confèrent la capacité de résister à des environnements évolutifs et/ou instables.

Selon les besoins applicatifs, les SMA utilisent des agents cognitifs rationnels, intentionnels, normatifs, etc. pouvant exhiber des comportements coopératifs ou compétitifs. Dans ce qui suit, le focus sera mis sur les agents rationnels et les agents intentionnels ; l'aspect normatif pouvant être intégré dans chacun de ces deux types d'agents.

3.2.1 Agents rationnels

Dans le domaine des SMA, la rationalité est étudiée d'un point de vue pratique, i.e. relativement à l'action qui lie plans, stratégies et intentions. On s'intéresse plus particulièrement aux actions rationnelles qui ont une forme logique et dont l'agent est responsable. Ainsi, un agent rationnel est défini comme un agent cognitif qui agit selon une rationalité donnée (par exemple, rationalité économique) en vue d'améliorer son état (par exemple se rapprocher

de son but). Cela suppose que l'agent est capable de mesurer sa performance et de savoir à quel moment effectuer cette mesure. La mesure de la rationalité dépend de quatre éléments :

- la mesure de performance qui définit le degré de succès d'un agent ;
- l'historique de la perception d'un agent ;
- la connaissance de l'agent à propos de son environnement ;
- les actions qu'un agent peut effectuer.

Dans [57], un **agent rationnel idéal** est défini ainsi : « pour tout historique possible de sa perception, cet agent effectuera l'action dont les connaissances à sa disposition permettent de penser qu'elle maximise sa performance ».

La conception d'agents rationnels, au sein des SMA, soulève de nombreux problèmes. Un des plus importants est celui de la définition de la rationalité d'un groupe d'agents, autonomes, souvent compétitifs. Les agents rationnels ont servi, essentiellement, à la conception et la simulation de systèmes compétitifs (par exemple économiques) dans les domaines de la négociation, la théorie des jeux ou de la formation de coalitions.

3.2.2 Agents intentionnels

Un **agent intentionnel** est un agent cognitif doté d'**attitudes intentionnelles** (ou **attitudes mentales**) telles que définies par Dennett [19] qui considère que l'intentionnalité est la stratégie permettant d'interpréter le comportement d'une entité (personne, animal, etc.) en la traitant comme un agent rationnel qui choisit ses actions en fonction de ses croyances et de ses désirs.

Dans le domaine des SMA, les agents intentionnels sont largement étudiés. Ils sont incarnés par les agents BDI (*Beliefs, Desires, Intentions*) [54] qui sont capables de raisonner sur leurs attitudes, de planifier et d'invoquer des plans selon la théorie du raisonnement pratique (*means-ends reasoning* comme défini dans [11]). Ces agents sont dotés de trois types d'**attitudes mentales** (dans la version de base du paradigme, qui a été ensuite étendue avec des normes ou des émotions) :

- **BELIEF** : la représentation (potentiellement fausse) de l'agent de son environnement, de lui-même et des autres agents ;
- **DESIRE** (parfois dénommés **GOAL** également) : les buts de l'agents, i.e. l'état du monde tel que l'agent souhaiterait qu'il soit ;
- **INTENTION** : les choix que l'agent s'engage (vis-à-vis de lui-même) à tenter de réaliser. Les intentions sont généralement associées à des **PLANS** (notamment des enchaînements d'actions) permettant de les atteindre.

Sur la base de ce socle commun, deux approches complémentaires se sont développées. Cette architecture a d'un côté été étudiée d'un point de vue formel : [54] et de nombreux autres chercheurs après eux ont décrit le système en logique modale avec une sémantique basée sur les mondes possibles ou plus précisément envisageables par l'agent. Chaque type d'attitude de l'agent est défini par l'ensemble des mondes que l'agent associe à cette attitude. De manière complémentaire, l'architecture BDI a été implémentée dans différents langages et architectures. Une des premières versions a été proposée par [55] dans une version simplifiée de l'architecture PRS (figure 2). Le fonctionnement de l'interpréteur est présenté dans la figure 3.

L'interpréteur BDI fonctionne comme suit (figure 3) :

- 1/ L'agent observe la queue d'événements et génère un ensemble d'*options* (i.e. de possibles nouvelles intentions).
- 2/ À partir de ces options, l'agent décide d'adopter un certain nombre de nouvelles intentions.

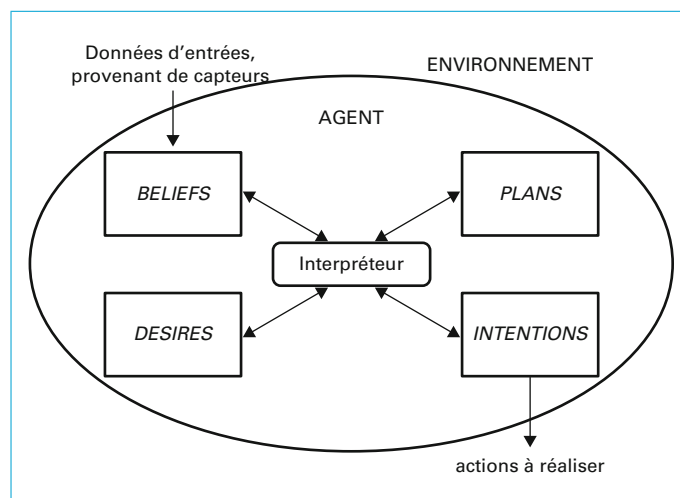


Figure 2 – Architecture du *Procedural Reasoning System* (PRS) [30], base des architectures BDI (traduit depuis [69])

```

/*Interpréteur d'agents BDI*/
Variables
b : croyances ; g : désirs/buts ; i : intentions ; eq : queue d'événements
Traitement
(b,g,i) := initialiser();
Répéter {
    options := générer_options(eq,b,g,i);
    selected := délibérer(options,b,g,i);
    i := selected U i;
    exécuter();
    eq := observer();
    b := mettre_à_jour_croyances(b,eq);
    (g,i) := supprimer_attitudes_accomplies(b,g,i);
    (g,i) := supprimer_attitudes_impossibles(b,g,i);
} Indéfiniment ;
    
```

Figure 3 – Algorithme d'un interpréteur d'agent BDI, adapté de [55]

3/ Si l'agent dispose d'une intention de réaliser une action atomique, il l'exécute.

4/ L'agent observe l'environnement et fait une mise à jour de la queue des événements (possiblement en fonction de ses attitudes mentales).

5/ L'agent met à jour ses attitudes mentales, en ajoutant de nouvelles croyances (provenant de l'observation de son environnement) ou en supprimant ses attitudes mentales réalisées ou impossibles à réaliser (en particulier ses intentions).

Ce modèle d'agents BDI a donné lieu à de nombreux outils (langages de programmation orientée agent, plateformes, etc.) opérationnels pour implanter des SMA dont certains ont été commercialisés comme la plateforme JACK. De nombreuses extensions des agents BDI ont également été proposées dans la littérature multi-agents comme l'agent BDI – HOA (*High Order Agent*) qui peut prendre en compte le contexte d'exécution et lancer plusieurs plans à la fois pour atteindre plusieurs buts simultanément [17]. En plus de leur utilisation pour des objectifs de résolution de problèmes, les agents BDI ont également été utilisés au sein de simulations multi-agents [1] pour représenter la complexité des comportements humains.

À retenir

- Les architectures d'agents peuvent être positionnées sur un continuum allant des agents **réactifs** aux agents **cognitifs**.
- Les **agents réactifs** ont un comportement réflexe lié à leurs perceptions et de type stimulus/action.
- Les **agents cognitifs** possèdent une représentation de leur environnement et des autres agents, ont des buts explicites et sont capables de planifier leur comportement, de mémoriser leurs actions passées, de communiquer, de négocier, etc.
- Parmi ces agents cognitifs, les **agents rationnels** tendent à choisir leurs actions afin de maximiser leur performance tandis que les **agents intentionnels** sont caractérisés par leur architecture à base d'attitudes mentales (Croyances, Buts et Intentions) qui vont diriger leurs actions.

4. Interactions entre agents

4.1 Interactions et langage de communication

Les **interactions** sont des actions réciproques que les agents effectuent pour se mettre en relation entre eux, s'influencer, coopérer ou s'organiser. Une interaction peut être réalisée directement via des messages que les agents échangent ou indirectement en agissant sur l'environnement ou un artefact.

L'interaction *via* l'environnement est une communication indirecte principalement utilisée par les agents réactifs. L'environnement y sert de médium d'interaction grâce à l'utilisation de marqueurs, de traces, de phéromones (modèles éthologiques)...

La communication directe est le plus souvent supportée par un langage de communication qui permet de transmettre de l'information, de demander ou rendre des services et indirectement, de changer l'état mental de l'agent destinataire d'un message. Des langages spécifiques de haut niveau, dits ACL (*Agent Communication Language*), ont été élaborés pour permettre la communication inter-agents. Les deux langages les plus populaires sont KQML (*Knowledge Query and Manipulation Language*) [27], et FIPA-ACL [28]. Ils reposent sur la théorie linguistique des actes de langage [2] qui considère que les communications ont également un effet performatif, c'est-à-dire constituent des actes (réfuter, affirmer, questionner, recommander...) cherchant à produire des effets sur le destinataire. Dans les ACL, le performatif accompagnant un message est explicitement exprimé.

Dans l'exemple présenté à la figure 4, un client demande aux vendeurs le type et le prix des vélos. Le contenu de la demande est exprimé dans le langage Prolog en se basant sur l'ontologie VEHICULES.

```
(ask-all          # le performatif
 : sender client   # émetteur du message
 : receiver vendeur # le récepteur du message
 : content « velo(Type, Prix) » # contenu
 : reply-with demande-velo # référence de la question
 : langage Prolog   # le langage d'expression du contenu
 : ontology VEHICULES # l'ontologie du domaine relative au contenu
)
```

Figure 4 – Le performatif ask-all en KQML

Les ACL facilitent la communication entre des agents hétérogènes puisque le langage d'expression du contenu d'un message est un paramètre, et qu'un message peut faire référence à une ontologie.

Pour structurer les communications entre agents, des protocoles d'interaction sont définis. À un plus haut niveau, ces protocoles vont permettre la collaboration entre agents et leur coordination. La sémantique des communications va aussi pouvoir s'exprimer au niveau de protocoles qui permettent de définir les enchaînements logiques entre les actes de communication.

4.2 Protocoles d'interaction

Les **protocoles d'interaction** permettent d'orchestrer et de contraindre les conversations entre agents pour atteindre un but et optimiser les échanges. Ainsi, une conversation est considérée comme une occurrence d'un Protocole d'Interaction. Le protocole d'appel d'offres « Contract Net », les différentes formes d'enchères, de vote ou de négociation sont des exemples de protocoles [29] [47] [59]. Un protocole se définit essentiellement par [34] :

- son but (par exemple vendre un objet dans un protocole d'enchères) ;
- les rôles que les agents peuvent jouer au sein d'une conversation (par exemple enchérisseur, vendeur) ;
- les types d'actes de communication autorisés pour chacun (par exemple attribuer l'objet, enchérir, payer...) ;
- l'état initial et final d'une conversation (par exemple respectivement : un objet est à vendre à un prix initial ; l'objet est vendu à un seul enchérisseur et le vendeur a été payé) ;
- les contraintes comportementales concernant le déroulement d'une conversation, c'est-à-dire les règles d'intervention (par exemple le prix doit être diminué si aucun enchérisseur n'est intéressé, la vente ne peut être conclue que si le paiement est réalisé...).

Chaque agent est libre de définir sa stratégie d'utilisation du protocole, et il peut mener plusieurs conversations simultanées conformes à un ou plusieurs protocoles.

Ces protocoles peuvent être spécifiés de manière répartie, c'est-à-dire qu'ils peuvent être dissous dans chaque rôle ou au contraire isolés et spécifiés comme un composant à part entière que les agents peuvent partager et utiliser à la manière d'une ontologie [32].

Des efforts de spécifications de ces protocoles ont été réalisés notamment avec la notation AUMML [3] par l'organisation FIPA. La figure 5 décrit la spécification du protocole « Contract Net » où un initiateur lance un appel d'offres (cfp) vers des participants (contractants). Les participants retenus font des propositions ou abandonnent. Des spécifications plus rigoureuses ont été élaborées avec le formalisme des réseaux de Petri qui permet l'analyse et la vérification de propriétés formelles de ces protocoles [23] [34].

4.3 Coordination

La présence dans un SMA d'agents (e.g. cognitifs, intentionnels, rationnels, etc.) plus ou moins autonomes et hétérogènes partageant le même environnement induit de nouveaux problèmes inhérents à la vision partielle des agents, la poursuite d'objectifs locaux et l'entrelacement des activités des agents. La coopération entre agents, en vue d'engendrer un comportement global cohérent du SMA, requiert des mécanismes élaborés de coordination afin d'éviter des conflits potentiels et de favoriser la synergie des activités des agents [22].

La **coordination** est un processus qui implique à un instant donné un ensemble d'agents du SMA et les aide à atteindre un état global cohérent i.e. où les états locaux des différents agents sont compatibles en termes d'objectifs, d'actions, d'utilisation et de partage des ressources, etc. Ainsi, tout au long de ce processus, les agents vont œuvrer pour éviter des situations conflictuelles (e.g. antagonisme, interblocage, famine, etc.) pouvant être engendrées par une mauvaise utilisation des ressources partagées, par

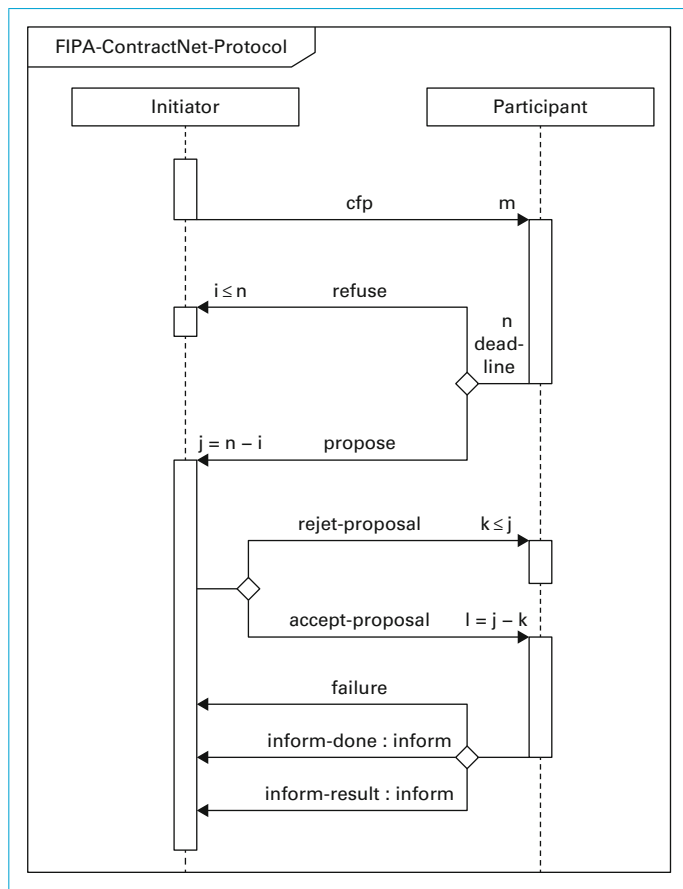


Figure 5 – Le protocole d'appel d'offres « contract net » modélisé avec la notation AUML selon FIPA [29]

l'existence de connaissances et de visions incompatibles, etc. Au-delà de la résolution de conflits potentiels, la coordination présente un intérêt majeur qui est celui de favoriser la synergie des agents en leur permettant de bénéficier de leurs capacités respectives, et de tirer profit des actions des uns et des autres. On parlera de situations d'interaction favorables (ou positives) entre agents.

4.3.1 Motivations de la coordination

Plus précisément, la coordination se révèle nécessaire pour plusieurs raisons liées les unes aux autres [32] [39] [49] :

■ La réconciliation de points de vue

Une des premières motivations de la coordination a été la prise en compte de points de vue conflictuels lors de la résolution distribuée de problèmes. Ce concept a été introduit par Kornfeld et Hewitt [40] dans leur travail « métaphore de la communauté scientifique » où la concurrence et la divergence de points de vue sont les clés de son succès. Il s'agit de pouvoir réconcilier des points de vue divergents et des intentions conflictuelles de différents agents pour garantir qu'ils agissent de manière cohérente lors d'une prise de décision ou dans le choix des actions futures, ainsi que lors de la recherche d'un compromis.

■ L'insuffisance individuelle des compétences, des ressources ou des informations

La capacité individuelle de chaque agent est souvent limitée et ne lui permet pas d'atteindre seul son objectif : c'est ce qui entraîne la nécessité de mettre en commun, de partager et d'échanger des

compétences, des ressources ou des informations. Cela exige de recourir à des mécanismes de répartition des tâches, à des règles d'accès et de partage des ressources, et à des protocoles d'échanges d'information entre agents.

■ L'interdépendance entre les actions, les plans ou les buts des agents

Il arrive que les tâches attribuées aux agents soient interdépendantes et nécessitent qu'ils coordonnent leurs actions, plans et buts. Par exemple, dans le domaine de l'acquisition coopérative d'information, des interdépendances entre les informations à retrouver nécessitent d'entrelacer et de synchroniser les actions de recherche et de transmission d'informations des différents agents informationnels impliqués [50].

■ La recherche d'efficacité

Il s'agit d'éviter une explosion combinatoire dans l'exploration d'alternatives, des dégradations des performances du système par un usage inconsidéré des communications, voire un gaspillage des ressources.

■ La satisfaction de contraintes non fonctionnelles

Le regroupement d'agents autour d'un objectif commun introduit également des contraintes non fonctionnelles à respecter par l'ensemble du système, telles que le temps limité dont les composants disposent pour rendre le service en un temps raisonnable, le coût financier du service...

4.3.2 Techniques de coordination

Il existe de nombreuses classifications des techniques de coordination [22] [25] [32] [49]. On considérera ici les six familles suivantes qui suffisent à donner une vue d'ensemble :

- la **Coordination par allocation de tâches**. Il s'agit d'allouer un ensemble de tâches à un groupe d'agents, éventuellement en optimisant des critères (coûts, délais...). Deux grandes familles de techniques existent : centralisées ou distribuées. Une première technique centralisée repose sur un agent « ordonnanceur » chargé de décomposer une tâche complexe en sous-tâches et de déléguer les sous-tâches à plusieurs agents, sélectionnés pour leur qualité de service (temps, coût, réputation...). Une deuxième technique centralisée se base sur un entremetteur (MatchMaker, Broker) qui met en relation des agents clients avec des agents fournisseurs de service. Dans le cas de l'allocation distribuée de tâches, on distingue plusieurs techniques dont l'allocation par réseau d'acointances, l'allocation basée sur des enchères ou par appel d'offres [59] ;

- la **Coordination basée sur des structures organisationnelles**. Elle consiste à structurer et réguler l'espace d'interaction d'un groupe d'agents par des mécanismes organisationnels qui déterminent les règles à respecter (conventions, normes), les relations sociales qu'ils entretiennent, les rôles qu'ils peuvent tenir et/ou les protocoles d'interaction qu'ils doivent suivre (cf. section 5). Il existe plusieurs formes d'organisation multi-agents [35] : hiérarchie, holarchie, coalition, équipe, fédération... ;

- la **Coordination basée sur des protocoles d'interaction**. Beaucoup de travaux ont été développés autour des protocoles pour deux raisons. Tout d'abord parce qu'ils permettent de coordonner les interactions et d'autre part car il s'agit d'un mécanisme qui se combine bien avec tous les autres. L'allocation de tâches, la négociation, la planification peuvent, à des étapes diverses, tirer profit du déploiement de protocoles ;

- la **Coordination d'actions par planification multi-agents**. Il s'agit de définir et d'accorder les plans de plusieurs agents en résolvant des éventuels conflits et en opérant des optimisations avant leur exécution. Il existe plusieurs formes de planification selon que chacune de ces étapes est réalisée par un ou plusieurs agents, que ces étapes sont séquentielles ou entrelacées, et qu'elles s'inscrivent dans un environnement statique ou dynamique. Dans une approche complètement centralisée, elle conduit à l'élaboration d'un plan global dont la définition, le contrôle de l'exécution est délégué à un

agent coordinateur. Dans une approche complètement distribuée, elle peut amener les agents à entrer dans une phase de négociation pour ajuster leur plan respectif et s'engager sur leur ordonnancement avant leur exécution distribuée ;

- **la négociation** a pour objectif de résoudre ou prévenir des conflits éventuels entre agents. Les conflits peuvent concerner l'accès aux ressources, des solutions différentes pour un même problème... En cas de conflits, les agents vont entrer dans un dialogue pour arriver ensemble à un compromis ;

- **la coordination via l'environnement ou des artefacts**. À partir de traces, d'informations ou par leur simple position dans l'environnement, les agents peuvent réagir en chaîne. De même, le mécanisme de tableau noir évoqué au § 1 intègre un mécanisme de contrôle agissant comme un coordinateur.

Pour conclure, il faut préciser qu'un SMA peut combiner plusieurs de ces techniques durant son exécution. Il existe des procédures pour sélectionner en temps réel la meilleure technique [10].

À retenir

- Un performatif est un énoncé qui constitue également un acte.
- Un Langage de Communication Agents (ACL) est un langage interagents à base de performatif. FIPA-ACL et KQML sont les ACL les plus connus.
- Un Protocole d'Interaction décrit le comportement d'une classe de conversations (votes, enchères...).
- Coordination : ensemble de mécanismes permettant le comportement cohérent et efficace des agents d'un SMA.

5. Organisation multi-agents

Le déploiement de SMA à grande échelle et dans un environnement ouvert (tel que celui du Web) exige une structuration de l'espace d'interaction et une régulation des agents. Dans un tel contexte, quand les agents nombreux, possiblement hétérogènes peuvent entrer et sortir du système, des problèmes d'interopérabilité, d'efficacité et de sécurité se posent : comment garantir la compréhension mutuelle des agents, faciliter leur coopération ou protéger le système contre les attaques d'agents malveillants ? Des domaines d'application critiques exigent également que le comportement global du système respecte des contraintes ou des propriétés générales (stabilité, respect des engagements, poursuite d'une stratégie) ou prenne en compte les caractéristiques organisationnelles du domaine (hiérarchie, procédure, réglementation) [66]. La structure organisationnelle d'un SMA peut être prédéfinie et spécifiée avant son déploiement. Elle peut ensuite demeurer statique ou être dynamique quand les agents sont en capacité de la faire évoluer pour s'adapter. Elle peut également être un phénomène émergent, résultant de l'auto-organisation des agents, mise en œuvre par leurs interactions et sans coordination centralisée.

La spécification d'une organisation permet d'introduire explicitement des concepts structurants qui offrent les bonnes abstractions pour appréhender les problèmes ci-dessus au bon niveau, structurer la conception et rendre plus efficace l'exécution. Ces concepts, inspirés d'organisations humaines, permettent de décrire cette société d'agents au niveau macro, indépendamment des caractéristiques internes de chaque agent participant (niveau micro).

Plus précisément, la spécification d'une organisation multi-agents se définit par les relations entre agents, leurs rôles et leurs modes d'interactions afin de répondre à un objectif commun [26]. Ces éléments vont faciliter le partage des connaissances, l'accès

aux ressources, la répartition des tâches et la coordination des actions.

L'organisation devient ainsi une dimension à part entière dans l'analyse et la conception des SMA. S'inspirant des sciences sociales, divers modèles de conception d'organisations artificielles ont été élaborés (AALAADIN, MOISE+) dont certains sont intégrés à des méthodes de conception (OMNI, SODA, Tropos, GaiaExOA, INGENIAS...) [37]. Ces modèles articulent divers concepts et se distinguent par les choix interdépendants effectués à plusieurs niveaux :

- **le type de structure**. Il existe toute une panoplie de structures, présentées et comparées dans [35] : fédérations, coalitions, groupes, équipes, marchés, holarchie ;

- **les règles de coordination**. Elles peuvent s'exprimer sous forme de normes, missions, réglementations, contrats ou de protocoles d'interaction incluant des rôles ;

- **les modalités déontiques**. Elles incluent les obligations, les engagements, les permissions, les violations ou les sanctions. Ces concepts peuvent être explicites pour préciser les règles de coordination ou implicitement formulées dans celles-ci ;

- **le cycle de vie de l'organisation**. Il peut comprendre selon les cas tout ou partie de ces étapes suivantes : conception, configuration, déploiement, changement, adaptation, évaluation mais aussi découverte [33].

En termes d'évaluation de structures organisationnelles, [31] propose un cadre théorique intéressant. Une structure organisationnelle est représentée par un graphe où les sommets sont les rôles, et les arcs des relations entre eux (contrôle, coordination et autorité). Ensuite, à partir de la structure du graphe, un ensemble de métriques (économie, connectivité, univocité) sont calculées pour déterminer des propriétés telles que l'efficacité, la robustesse ou la flexibilité d'une organisation.

Exemple de modèle organisationnel : Un des modèles organisationnels les plus simples et néanmoins efficaces est le modèle AGR connu aussi sous le nom d'AALAADIN [26] (figure 6) qui est basé sur les concepts d'agent, groupe et rôle (AGR). De plus, des contraintes et des protocoles inter-rôles contraignent et fixent les règles d'interaction des agents au sein d'un groupe. Un agent peut jouer plusieurs rôles au sein de différents groupes. Un groupe est un espace d'interaction entre agents créé dynamiquement pour un objectif déterminé. Un agent ne peut communiquer avec un autre que s'ils appartiennent tous deux à un même groupe. Le manager du groupe est un rôle spécifique, rempli par l'agent qui a créé le groupe. Pour qu'un agent puisse jouer un certain rôle, l'autorisation du manager de groupe est requise. Il est aussi à noter que ce modèle est implémenté dans la plateforme de développement Madkit (<http://www.madkit.net>). En guise d'illustration, un groupe peut représenter une salle des ventes aux enchères créée par un agent vendeur et accueillant des agents acheteurs. Dans un même groupe, un vendeur ne peut pas être un acheteur. Un protocole d'enchères fixe le comportement de l'acheteur et du vendeur et garantit que la transaction est fiable. Plusieurs groupes peuvent évoluer concurrentement, correspondant à des salles de ventes différentes d'un même marché.

À retenir

- Une organisation multi-agents spécifie des relations entre agents, leurs rôles et leurs modes d'interaction afin de répondre à un objectif commun.
- La spécification de l'organisation d'un SMA structure la conception, rend plus efficace l'exécution, simplifie l'interopérabilité et facilite la sécurité.
- Une organisation peut être prédéfinie ou émergente.
- Une organisation multi-agents peut être explicitement décrite à l'aide de modèles comme Moise+ ou AALAADIN.

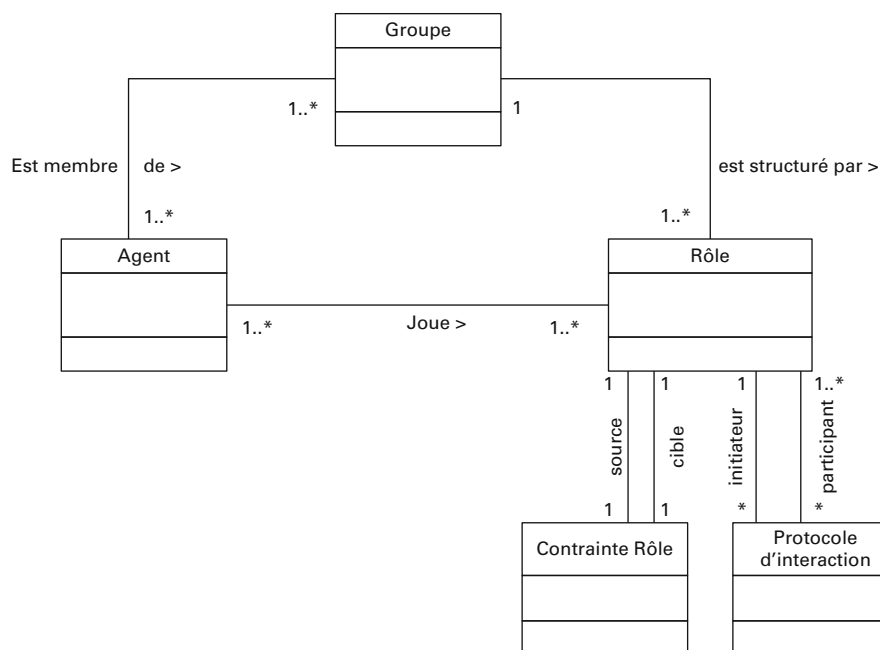


Figure 6 – Diagramme de classes UML du méta-modèle AGR [26]

6. Apprentissage en univers multi-agents

Il n'est pas toujours possible de définir complètement et précisément, en phase de conception, l'environnement dans lequel un agent évoluera, ni les règles de coordination d'un SMA. Il est aussi difficile d'optimiser a priori les comportements de ses agents du fait du caractère ouvert, évolutif et imprédictible de l'environnement et de la complexité des situations qu'ils peuvent être amenés à rencontrer. **L'apprentissage automatique** permet à un agent de découvrir et de s'adapter progressivement à son environnement, d'être plus efficace ou d'apprendre à se coordonner avec ses accointances pour réaliser une tâche complexe. L'exemple d'une équipe de robots footballeurs est un cas où l'apprentissage permet à une équipe d'établir des stratégies, à chaque coéquipier d'apprendre à tenir un rôle (gardien, défenseur, attaquant, milieu) au service de l'objectif commun (gagner un match) et d'adapter le comportement collectif selon l'équipe adverse dont le comportement et l'organisation ne sont pas connus a priori.

Il existe trois grandes approches d'apprentissage : **l'apprentissage supervisé** qui consiste à apprendre à prédire une réponse (sortie désirée) à partir d'une observation et en se basant sur une base d'exemples sous forme de couples (observation, sortie désirée). **L'apprentissage non supervisé** consiste à catégoriser les entrées pour définir des classes. Enfin, **l'apprentissage par renforcement** permet à un agent d'apprendre à partir de ses succès et ses échecs, auxquels est attribuée une évaluation (récompense ou pénalité). Cette dernière approche est la plus adaptée et étudiée dans le cadre des SMA et c'est celle que l'on développera dans la suite.

Les **jeux stochastiques** (ou Jeux de Markov) constituent un des cadres théoriques de référence pour l'apprentissage par renforcement en univers multi-agents [43]. Il combine les modèles de décisions de Markov et les jeux matriciels de la théorie des jeux qui

permettent de déterminer des équilibres (situations stables). Un jeu stochastique est défini comme un n-uplet (P, S, A, R, T) où :

- P est un ensemble de n agents ;
- S est l'ensemble discret de k états de l'environnement ;
- $A = A_1 \times A_2 \times \dots \times A_i \times \dots \times A_n$, l'ensemble des actions conjointes des agents. A_i est l'ensemble discret des actions disponibles pour l'agent i ;
- $R = \{R_i : i=1\dots n\}$ ensemble des fonctions de renforcement. $R_i : S \times A \rightarrow \text{Réels}$ est la fonction de renforcement de l'agent i . Elle dépend de l'action conjointe des agents dans un certain état ;
- $T : S \times A \times S \rightarrow [0,1]$ est une fonction de transition du système en réaction aux actions conjointes des agents. Elle indique la probabilité de passer d'un état $s \in S$ à un état $s' \in S$ en exécutant l'action conjointe $a \in A$.

On remarquera que R , T et l'état suivant dépendent des actions conjointes des différents agents. Le comportement d'un agent est aussi déterminé par une politique, fonction qui le guide dans le choix de l'action à entreprendre en fonction de son état. La difficulté est alors pour l'agent d'adopter une politique de qualité, c'est-à-dire lui permettant de maximiser ses futurs gains, sachant qu'elle dépend aussi des politiques de ses accointances. Dans ces conditions, il peut lui être difficile d'assurer la convergence vers un comportement optimal, et il est plus facile de chercher un équilibre au sens de Nash.

Il existe d'autres cadres théoriques, plus réalistes, dans lesquels notamment les agents n'ont qu'une perception partielle de leur environnement (exemple : Dec-POMDP) ou qui prennent en compte l'évolution dans le temps des réactions de l'environnement (propriété de non-stationnarité).

Plus généralement, il existe plusieurs méthodes et algorithmes d'apprentissage qui se différencient selon plusieurs critères [65] :

- la compatibilité des buts qui conduit à des apprentissages coopératifs, compétitifs ou mixtes. Dans le cas coopératif, les agents ont le même but (par exemple, gagner le match pour les membres

d'une même équipe) et la même fonction de récompense. Ils cherchent alors à maximiser leur récompense en tant que groupe. Dans le cas compétitif, les agents ont des buts conflictuels et chacun cherche à réaliser son but au détriment des autres (par exemple, 2 adversaires au jeu d'échecs). Les agents cherchent à maximiser leur récompense individuelle. Le cas mixte consiste à combiner des comportements coopératifs et compétitifs selon les situations ;

- la distribution de l'apprentissage. On distingue principalement l'apprentissage en équipe où un agent seul est chargé de l'apprentissage au profil d'un groupe d'agents, de l'apprentissage concurrent où chaque agent exécute son processus d'apprentissage ;
- le degré de connaissance (« *awareness* ») que les agents ont de leurs accointances et de leur processus d'apprentissage ;
- le degré d'homogénéité des algorithmes d'apprentissage des agents ;
- le degré de perception de l'environnement.

Plus récemment, des techniques d'apprentissage profond ont été combinées aux méthodes par renforcement pour traiter le problème de l'explosion des espaces de recherche due à la taille des ensembles d'actions conjointes et des états de l'environnement. On trouvera dans [45] une comparaison récente des différentes méthodes multi-agents d'apprentissage par renforcement profond.

[52] présente une revue exhaustive des domaines d'application réels et d'études de cas. Parmi ces domaines, on compte le contrôle distribué (trafic aérien, réseaux électriques), la négociation en ligne et l'ordonnancement dans les chaînes logistiques... La robotique collective (footballeurs, explorateurs, observateurs) est un des domaines les plus étudiés en recherche.

À retenir

- L'apprentissage automatique permet à un agent de découvrir et de s'adapter progressivement à son environnement, d'être plus efficace ou d'apprendre à se coordonner avec ses accointances pour réaliser une tâche complexe.
- Trois grandes techniques d'apprentissage sont exploitées : supervisé, non supervisé et par renforcement.
- Au niveau collectif, l'apprentissage peut être coopératif, compétitif ou mixte.
- Les jeux stochastiques constituent un des cadres théoriques de référence pour l'apprentissage par renforcement en univers multi-agents.

7. Ingénierie des SMA : méthodes et outils

En termes d'ingénierie des Systèmes Multi-Agents, on peut noter une profusion d'approches proposées que ce soit en termes de méthodologies, d'outils pour développer des systèmes multi-agents ou de modèles à base d'agents.

7.1 Méthodologies

La méthodologie la plus employée en ingénierie des systèmes multi-agents est la méthodologie GAIA [71]. À partir d'un ensemble de prérequis, une phase d'analyse s'intéresse à diviser le système en termes de sous-organisation débouchant sur la proposition de modèles d'organisation, d'interaction et d'environnement raffinés par la suite. C'est seulement dans la dernière phase de conception détaillée que la notion d'agents apparaît. D'autres méthodologies comme PROMETHEUS [68] (ou TROPOS [14]) se focalisent plutôt sur une approche de SMA à base d'agents *intelligents*. Après une phase d'identification des fonctionnalités, entrées, sorties et données disponibles, la méthodologie se concentre sur l'identification

des agents et de leurs interactions, avant une dernière étape de conception détaillée se concentrant sur les états internes des agents (et en particulier les aspects cognitifs). La méthodologie INGENIAS [53], de son côté, se base sur la méthode classique de génie logiciel orienté objet du processus unifié (*Unified Process*). Elle se base donc sur une approche incrémentale utilisant différents cas d'études et méta-modèles décrivant les principaux éléments du SMA (Agent, Interaction, Organisation, Environnement et buts ou tâches). Enfin la méthode ADELFE [5] est dédiée au développement de systèmes aux propriétés émergentes et plus particulièrement aux systèmes auto-adaptatifs suivant la théorie des AMAS. La spécificité de cette méthode est de s'attacher à identifier et résoudre, au moyen de la collaboration entre agents, les situations non-coopératives du système.

7.2 Outils

Pour présenter un aperçu des principaux outils de développement de systèmes multi-agents, une séparation sera faite entre outils de développement de systèmes multi-agents et plateformes de modélisation et simulation à base d'agents. Cette séparation provient à la fois de communautés, d'objectifs, de publics visés et d'outils disponibles différents. En particulier, les premiers ont plutôt vocation à être utilisés par des informaticiens pour développer des applications générales tandis que les seconds sont plutôt développés pour des experts dans d'autres domaines voulant développer un modèle à base d'agents sur leur question et thème d'étude. Ils disposent ainsi d'un ensemble beaucoup plus élaboré d'outils de visualisation et d'exploration des simulations.

7.2.1 Outils de développement de systèmes multi-agents

Comme présenté dans [6], les outils de développement de SMA incluent des plateformes d'exécutions d'agents et de systèmes multi-agents (dont notamment JADE ou IODA [42]), des langages de programmation d'agents (souvent cognitifs comme Jason, Jadex [13]) et des frameworks combinant plusieurs langages et approches (par exemple JaCaMo). Par ailleurs, on observe souvent le développement d'outils se concentrant sur un seul aspect des systèmes multi-agents. Ainsi on recense nombre d'outils se concentrant sur l'aspect cognitif (et notamment BDI) des agents (Jason, Jadex...), l'aspect organisation (MOISE [36] ou MadKit avec l'implémentation du modèle AGR), interactions (IODA), environnement (CARtAgO [56]) ou communication et distribution (JADE).

Parmi les plateformes d'exécution de systèmes multi-agents, JADE [4] est celle qui s'est imposée. Elle a, en effet, été, dès sa conception, la plateforme compatible avec le standard FIPA-ACL (que ce soit en termes de la librairie des actes de langages, des protocoles d'interaction ou des spécifications de la structure des messages ou de leur transport) : elle gère ainsi l'exécution des agents et en particulier la couche communication, que les agents soient sur le même ordinateur ou distribués sur un réseau. WADE [16] est une extension de JADE permettant d'exécuter des tâches définies selon un formalisme de *workflow*.

Comme présenté en Section 3, la principale architecture d'agents cognitifs est l'architecture *Belief-Desire-Intention* (BDI). [30] proposa l'architecture « *Procedural Reasoning System* » (PRS). Dix ans plus tard, [18] proposèrent le langage dédié AgentSpeak. Un des langages de programmation les plus utilisés dans ce domaine, étonnamment AgentSpeak, est Jason [9]. Il se base sur les principaux concepts du modèle BDI (*Beliefs, Desires, Intentions, and actions*) et les actes de langage pour communiquer entre agents. Les étapes du cycle de délibération peuvent également être redéfinies.

Comme vu précédemment, les différents développements d'outils ont souvent été réalisés en parallèle et en concurrence. Une exception notable et une approche intéressante sont promues par les auteurs de JaCaMo [7]. JaCaMo est en effet un *framework* de développement de systèmes multi-agents basé sur l'intégration

de Jason (en ce qui concerne l'architecture des Agents), CArtoG (qui externalise tout ce qui concerne l'Environnement de l'agent et en particulier les opérations et les événements observables, qui sont par exemple liés aux croyances de l'agent) et MOISE (regroupant les notions de rôle, groupe ou normes de la partie Organisation du SMA ; les normes pouvant ainsi influencer les buts de l'agent). Enfin ce méta-modèle est complété par une partie Interaction (et en particulier communication) pour fournir le panorama complet des différents concepts permettant d'implémenter un SMA.

Enfin cette idée d'intégration d'outils existants est également au cœur de l'approche proposée par [58] qui fournit un *middleware* permettant de coupler n'importe quel moteur d'agents BDI (par exemple JASON...) au sein de n'importe quelle plateforme de modélisation et simulation à base d'agents afin d'intégrer des agents cognitifs dans ces modèles.

7.2.2 Plateforme de modélisation et simulation à base d'agents

En parallèle des outils de développement de systèmes multi-agents, de nombreuses plateformes de modélisation et simulation à base d'agents [64] se sont développées. Ces outils (dont l'exemple prototypique est Netlogo [63]) visent à permettre à tout chercheur (sans expertise en informatique) de développer un modèle à base d'agents dans son domaine d'expertise, de le visualiser et de l'explorer. Ces outils, en plus de favoriser l'acceptation de ce paradigme de modélisation en l'ouvrant à des non-informaticiens, permettent également le partage et les échanges autour de ces modèles.

Par ailleurs, ces outils se sont développés pour intégrer les principaux composants nécessaires au développement des Systèmes Multi-Agents (communication, raisonnement ou prise de décision) et peuvent de ce fait être utilisés comme des outils de prototypage de systèmes multi-agents et donc comme support à l'enseignement des Systèmes Multi-Agents. En ce sens, [41] présente un état de l'art intégrant à la fois des outils de développement de systèmes multi-agents et des plateformes de modélisation.

Parmi les plateformes *open source* et suffisamment génériques pour modéliser tout type de système et de phénomène, deux principales approches ont émergé, illustrées par les deux principales plateformes. D'un côté, NetLogo [63] fournit un langage simple, dédié au développement de modèles, bénéficiant d'une courbe d'apprentissage très rapide. Cette plateforme est bien adaptée pour développer rapidement des modèles simples et de petite taille, mais reste limitée pour les modèles plus complexes. De l'autre côté, les développeurs de Repast Symphony [48] ont choisi de fournir une plateforme où le modélisateur doit développer son modèle dans le langage générique Java. Ce choix peut rendre l'outil plus ardu d'accès pour des non-informaticiens, mais le rend plus puissant car cela permet aux développeurs d'avoir un accès direct à toutes les bibliothèques Java existantes.

Dès 2007, le développement de la plateforme GAMA [62] a eu pour objectif de fournir la puissance d'expressivité de Repast Symphony (notamment en intégrant de manière intuitive et directe des données géographiques raster et vectorielles dans les modèles) tout en fournissant un langage de modélisation GAML dédié. De plus, GAMA permet de développer des agents situés et réactifs, intègre des extensions pour permettre aux agents de communiquer en utilisant les actes de langages et protocoles d'interaction du standard FIPA et de s'échanger des messages à distance ou de communiquer avec un serveur distant (par exemple MQTT). L'architecture de base des agents peut être soit une architecture à base de réflexes ou de tâches, d'automates à états finis, soit une architecture de raisonnement BDI. Par ailleurs, des extensions permettent aux agents de raisonner et de prendre des décisions à base de logique floue, réseau bayésien ou d'algorithme de décision multicritère. Une originalité du méta-modèle de GAMA est que tout est agent dans un modèle GAMA, les agents du modèle, mais aussi ce qui exécute ce modèle à savoir

les simulations et les plans d'expériences. Ceci est possible car tout modèle GAMA est par nature **multiniveau** [67] : tout agent est « hébergé » par un macro-agent (qui gère son exécution et dont sa forme géométrique constitue l'environnement de ses micro-agents) et peut héberger d'autres micro-agents. Une conséquence naturelle a été le développement d'une approche de **co-modélisation** [20] : comme toute simulation est un agent, on peut donc créer un agent créant plusieurs simulations et les couplant, et en particulier en gérant leurs interactions, de la même manière que deux agents d'un modèle interagissent.

Pour conclure, une tendance qui semble se dégager concernant les futurs développements à la fois des outils de développement ou des plateformes de modélisation et de simulation est l'intégration des humains dans les systèmes et plus généralement de données en temps réel depuis le monde réel. Cela nécessitera notamment des améliorations en termes de communication et de tendre vers des systèmes ouverts [51], menant ainsi au développement de « *live simulations* » [60] et de jumeau numérique (*digital twin*).

À retenir

- Les avancées théoriques des systèmes multi-agents se sont accompagnées de développements méthodologiques (pour concevoir une application basée sur le paradigme Multi-Agents) ou d'outils permettant de développer ces applications.
- Les plateformes de modélisation et de simulation à base d'agents permettent de développer rapidement des modèles spatialisés, nourris en données hétérogènes et de visualiser l'évolution de la simulation au niveau des agents individuels ou d'indicateurs macroscopiques.

8. Conclusion

Les systèmes multi-agents sont aujourd'hui arrivés à maturité avec un large domaine d'application et des méthodes et outils permettant leur ingénierie. Nous avons présenté ici les concepts essentiels pour permettre au lecteur de comprendre les fondements conceptuels des SMA et débiter leur apprentissage pratique méthodiquement. Nous avons en particulier mis en avant les aspects cognitifs et sociaux qui font des SMA un paradigme original de conception et de développement d'applications distribuées exhibant un comportement collectif intelligent. Parmi les thèmes en plein essor, nous avons choisi de détailler l'apprentissage par renforcement en univers multi-agents qui ouvre des perspectives prometteuses. Si les avancées sont nombreuses, de nouveaux défis ambitieux restent à relever, comme ceux de l'éthique, la responsabilité ou l'explication de décisions collectives prises par un groupe d'agents. Ces défis sont rendus difficiles par le caractère distribué du contrôle d'un SMA et sa possible auto-organisation (coalition, fédération...) qui échappent à son concepteur.

9. Sigles, notations et symboles

Symbole	Description	Unité
AAMAS	Autonomous Agents and Multi-Agent Systems	–
ACL	Agent Communication Language	–

Symbole	Description	Unité
ADELFE	Atelier pour le DÉveloppement de Logiciels à Fonctionnalité Émergente	–
AGR	Agent, Groupe, Rôle	–
AUML	Agent UML	–
BDI	<i>Beliefs, Desires, Intentions</i>	–
CARTAgO	<i>Common ARTifact infrastructure for AGents Open environments</i>	–
FIPA	<i>Foundation for Intelligent Physical Agents</i>	–
FIPA-ACL	<i>Foundation for Intelligent Physical Agents – Agent Communication Language</i>	–
GaiaExOA	<i>Gaia Extended with Organizational Abstractions</i>	–
GAMA	<i>GIS Agent-based Modeling Architecture</i>	–
GAML	<i>GAMA Modeling Language</i>	–
IA	Intelligence Artificielle	–
IAD	Intelligence Artificielle Distribuée	–
IJAOS	<i>International Journal of Agent-Oriented Software Engineering</i>	–
IJCAI	<i>International Joint Conferences on Artificial Intelligence</i>	–
INGENIAS	<i>Engineering for Software Agents</i>	–

Symbole	Description	Unité
IODA	<i>Interaction-Oriented Design of Agent simulations</i>	–
JAAMAS	<i>Journal of Autonomous Agents and Multi-Agent Systems</i>	–
JaCaMo	JADE CARTAgO MOISE	–
JADE	<i>Java Agent DEvelopment Framework</i>	–
JASSS	<i>Journal of Artificial Societies and Social Simulation</i>	–
KQML	<i>Knowledge Query and Manipulation Language</i>	–
MaDKit	<i>The Multi-agent Development Kit</i>	–
MAOP	<i>Multi-Agent Oriented Programming</i>	–
MOISE	<i>Model of Organization for multi-agent Systems</i>	–
MQTT	<i>Message Queuing Telemetry Transport</i>	–
PRS	<i>Procedural Reasoning System</i>	–
SMA	Système Multi-Agents	–
SoS	<i>System of Systems</i>	–
UML	<i>Unified Modeling Language</i>	–
WADE	<i>Workflows and Agent Development Environment</i>	–

Systèmes multi-agents

par **Frédéric AMBLARD**

Professeur, Université Toulouse 1 Capitole,
Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505, Toulouse, France

Amal EI FALLAH-SEGHROUCHNI

Professeure, Sorbonne Université,
Laboratoire d'Informatique de Paris 6, UMR CNRS 7606, Paris, France

Benoît GAUDOU

Maître de conférences, Université Toulouse 1 Capitole,
Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505, Toulouse, France

et **Chihab HANACHI**

Professeur, Université Toulouse 1 Capitole,
Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505, Toulouse, France

Sources bibliographiques

- [1] ADAM (C.) et GAUDOU (B.). – *BDI agents in social simulations: a survey*. Knowledge Engineering Review, vol. 31, no. 3, pp. 207-238 (2016).
- [2] AUSTIN (J.L.). – *How to do things with words*. Oxford university press (1962).
- [3] BAUER (B.), MÜLLER (J.P.) et ODELL (J.). – *Agent UML: A Formalism for Specifying Multiagent Interaction*. In: P. CIANCARINI and M. WOOLDRIDGE, eds. Agent-Oriented Software Engineering. Springer, p. 91-103 (2001).
- [4] BELLIFEMINE (F.L.), CAIRE (G.) et GREENWOOD (D.). – *Developing multi-agent systems with JADE*. John Wiley & Sons (2007).
- [5] BERNON (C.), GLEIZES (M.-P.), PEYRUQUEOU (S.) et al. – *ADELFE: A methodology for adaptive multi-agent systems engineering*. In: International Workshop on Engineering Societies in the Agents World. Springer, p. 156-169 (2002).
- [6] BOISSIER (O.). – *Outils et langages de programmation d'applications multi-agents*. PDIA (2017).
- [7] BOISSIER (O.), BORDINI (R.H.), HUBNER (J.) et al. – *Multi-agent oriented programming: programming multi-agent systems using JaCaMo*. MIT Press (2020).
- [8] BORDINI (R.H.), EL FALLAH SEGHRUCHNI (A.), HINDRIKS (K.) et al. – *Agent programming in the cognitive era*. Autonomous Agents and Multi-Agent Systems, vol. 34, pp. 1-31 (2020).
- [9] BORDINI (R.H.), HÜBNER (J.F.) et WOOLDRIDGE (M.). – *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons (2007).
- [10] BOURNE (R.A.), EXCELENTE-TOLEDO (C.B.) et JENNINGS (N.R.). – *Run-time selection of coordination mechanisms in multi-agent systems*. In: In Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000). IOS Press, p. 348-352 (2000).
- [11] BRATMAN (M.). – *Intention, plans, and practical reason*. Harvard University Press Cambridge, MA (1987).
- [12] BRATMAN (M.E.), ISRAEL (D.J.) et POLLACK (M.E.). – *Plans and resource-bounded practical reasoning*. Computational intelligence, vol. 4, no. 3, pp. 349-355 (1988).
- [13] BRAUBACH (L.), POKAHR (A.) et LAMERSDORF (W.). – *Jadex: A BDI-agent system combining middleware and reasoning*. In: Software agent-based applications, platforms and development kits. Springer, p. 143-168 (2005).
- [14] BRESCIANI (P.), PERINI (A.), GIORGINI (P.) et al. – *Tropos: An agent-oriented software development methodology*. Autonomous Agents and Multi-Agent Systems, vol. 8, no. 3, pp. 203-236 (2004).
- [15] BROOKS (R.A.). – *Intelligence without representation*. Artificial intelligence, vol. 47, nos. 1-3, pp. 139-159 (1991).
- [16] CAIRE (G.), GOTTA(D.) et BANZI (M.). – *Wade: a software platform to develop mission critical applications exploiting agents and workflows*. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track, p. 29-36 (2008).
- [17] CHAOUICHE (A.-C.), EL FALLAH-SEGHROUCHNI (A.), ILIÉ (J.-M.) et al. – *A higher-order agent model with contextual planning management for ambient systems*. In: Transactions on Computational Collective Intelligence XVI. Springer, p. 146-169 (2014).
- [18] D'INVERNO (M.) et LUCK (M.). – *Engineering AgentSpeak (L): A formal computational model*. Journal of Logic and Computation, vol. 8, no. 3, pp. 233-260 (1998).
- [19] DENNETT (D.C.). – *The intentional stance*. MIT Press, (1989).
- [20] DROGOUL (A.), HUYNH (N.Q.) et TRUONG (Q.C.). – *Coupling environmental, social and economic models to understand land-use change dynamics in the Mekong Delta*. frontiers in environmental science, vol. 4, p. 19 (2016).
- [21] DURFEE (E.H.) et ROSENSCHEIN (J.S.). – *Distributed problem solving and multi-agent systems: Comparisons and examples*. In: Proceedings of the Thirteenth International Distributed Artificial Intelligence Workshop, p. 94-104 (1994).
- [22] EL FALLAH-SEGHROUCHNI (A.). – *Modèles de coordination d'agents cognitifs*. In: J.-P. BRIOT and Y. DEMAIZEAU, eds. Principes et architecture des systèmes multi-agents. Hermes Science Publications, Informatique et Systèmes d'Information, p. 139-176 (2001).
- [23] EL FALLAH-SEGHROUCHNI (A.), HADDAD (S.) et MAZOUZI (H.). – *Protocol engineering for multi-agent interaction*. In: European Workshop on Modelling Autonomous Agents in a Multi-Agent World. Springer, p. 89-101 (1999).
- [24] ERMAN (L.D.), HAYES-ROTH (F.), LESSER (V.R.) et al. – *The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty*. ACM Computing Surveys (CSUR), vol. 12, no. 2, pp. 213-253 (1980).
- [25] FERBER (J.). – *Les systèmes multi-agents: vers une intelligence collective*. InterEditions.
- [26] FERBER (J.), GUTKNECHT (O.) et MICHEL (F.). – *From agents to organizations: an organizational view of multi-agent systems*. In: International workshop on agent-oriented software engineering. Springer, 2003, p. 214-230 (1997).
- [27] FININ (T.), FRITZSON (R.), MCKAY (D.) et al. – *KQML as an agent communication language*. In: Proceedings of the third international conference on Information and knowledge management. p. 456-463 (1994).
- [28] FIPA. – *FIPA Communicative Act Library Specification*. <http://www.fipa.org/specs/fipa00037/>. Foundation for Intelligent Physical Agents (2002).

- [29] FIPA. – *FIPA Contract Net Interaction Protocol Specification*. <http://www.fipa.org/specs/fipa00029/>. Foundation for Intelligent Physical Agents (2002).
- [30] GEORGEFF (M.P.) et LANSKY (A.L.). – *Reactive reasoning and planning*. In: AAAI., p. 677-682 (1987).
- [31] GROSSI (D.), DIGNUM (F.), DIGNUM (V.) et al. – *Structural evaluation of agent organizations*. In: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, p. 1110-1112 (2006).
- [32] HANACHI (C.). – *La coordination dans les systèmes d'information agents*. B.m., Habilitation à Diriger des Recherches, Université Toulouse 1 Capitole (2003).
- [33] HANACHI (C.), GAALOUL (W.) et MONDI (R.). – *Performative-based mining of workflow organizational structures*. In: International Conference on Electronic Commerce and Web Technologies. Springer, p. 63-75 (2012).
- [34] HANACHI (C.) et SIBERTIN-BLANC (C.). – *Protocol moderators as active middle-agents in multi-agent systems*. Autonomous Agents and Multi-Agent Systems, vol. 8, no. 2, pp. 131-164 (2004).
- [35] HORLING (B.) et LESSER (V.). – *A survey of multi-agent organizational paradigms*. Knowledge Engineering Review., vol. 19, no. 4, pp. 281-316 (2004).
- [36] HÜBNER (J.F.), BOISSIER (O.), KITIO (R.) et al. – *Instrumenting multi-agent organisations with organisational artifacts and agents*. Autonomous agents and multi-agent systems, vol. 20, no. 3, pp. 369-400 (2010).
- [37] ISERN (D.), SÁNCHEZ (D.) et MORENO (A.). – *Organizational structures supported by agent-oriented methodologies*. Journal of Systems and Software, vol. 84, no. 2, pp. 169-184 (2011).
- [38] JENNINGS (N.R.), SYCARA (K.) et WOOLDRIDGE (M.). – *A roadmap of agent research and development*. Autonomous agents and multi-agent systems, vol. 1, no. 1, pp. 7-38 (1998).
- [39] JENNINGS (N.R.). – *Coordination Techniques for Distributed Artificial Intelligence*. In: Foundations of Distributed Artificial Intelligence. USA: John Wiley & Sons, Inc., p. 187-210. ISBN 0-471-00675-0 (1996).
- [40] KORNFIELD (W.A.) et HEWITT (C.E.). – *The scientific community metaphor*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 11, no. 1, pp. 24-33 (1981).
- [41] KRAVARI (K.) et BASSILIADES (N.). – *A survey of agent platforms*. Journal of Artificial Societies and Social Simulation, vol. 18, no. 1, p. 11 (2015).
- [42] KUBERA (Y.), MATHIEU (P.) et PICAULT (S.). – *IODA: an interaction-oriented approach for multi-agent based simulations*. Autonomous Agents and Multi-Agent Systems, vol. 23, no. 3, pp. 303-343 (2011).
- [43] LITTMAN (M.L.). – *Markov games as a framework for multi-agent reinforcement learning*. In: Machine learning proceedings 1994. Elsevier, p. 157-163 (1994).
- [44] MAIER (M.W.). – *Architecting principles for systems-of-systems*. Systems Engineering: The Journal of the International Council on Systems Engineering, vol. 1, no. 4, pp. 267-284 (1998).
- [45] NGUYEN (T.T.), NGUYEN (N.D.) et NAHAVANDI (S.). – *Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications*. IEEE transactions on cybernetics, vol. 50, no. 9, pp. 3826-3839 (2020).
- [46] NIL (H.P.). – *Blackboard systems: Blackboard application systems, blackboard systems from a knowledge engineering perspective*. AI magazine, vol. 7, no. 3, pp. 82-106 (1986).
- [47] NORIEGA (P.) et SIERRA (C.). – *Auctions and multi-agent systems*. In: M. KLUSCH, ed. Intelligent Information Agents. Springer, p. 153-175 (1999).
- [48] NORTH (M.J.), COLLIER (N.T.), OZIK (J.) et al. – *Complex adaptive systems modeling with Repast Symphony*. Complex adaptive systems modeling, vol. 1, no. 1, pp. 1-26 (2013).
- [49] NWANA (H.S.), LEE (L.) et JENNINGS (N.R.). – *Co-ordination in multi-agent systems*. In: H.S. NWANA and N. AZARMIN, eds. Software Agents and Soft Computing Towards Enhancing Machine Intelligence. Springer-Verlag, Berlin, LNAI 1198, p. 42-58 (1997).
- [50] OATES (T.), PRASAD (M.N.) et LESSER (V.R.). – *Cooperative information-gathering: a distributed problem-solving approach*. IEE Proceedings-Software, vol. 144, no. 1, pp. 72-88 (1997).
- [51] PALANCA (J.), TERRASA (A.), JULIAN (V.) et al. – *SPADE 3: Supporting the New Generation of Multi-Agent Systems*. IEEE Access., vol. 8, pp. 182537-182549 (2020).
- [52] PANAIT (L.) et LUKE (S.). – *Cooperative multi-agent learning: The state of the art*. Autonomous agents and multi-agent systems, vol. 11, no. 3, pp. 387-434 (2005).
- [53] PAVÓN (J.), GÓMEZ-SANZ (J.J.) et FUENTES (R.). – *The INGENIAS methodology and tools*. In: Agent-oriented methodologies. IGI Global, p. 236-276 (2005).
- [54] RAO (A.) et GEORGEFF (M.). – *Modeling rational agents within a BDI-architecture*. In: Proc. 2nd Int. Conf. on Knowledge Representation and Reasoning (KR'91). Morgan Kaufmann. of, p. 473-484 (1991).
- [55] RAO (A.S.) et GEORGEFF (M.P.). – *BDI agents: From theory to practice*. In: Proceedings of the first International Conference on Multi-Agent Systems, p. 312-319 (1995).
- [56] RICCI (A.), PIUNTI (M.), VIROLI (M.) et al. – *Environment programming in CArtoG*. In: Multi-agent programming. Springer, p. 259-288 (2009).
- [57] RUSSELL (S.) et NORVIG (P.). – *Artificial Intelligence: A modern approach*. Prentice Hall, NJ, USA (1995).
- [58] SINGH (D.), PADGHAM (L.) et LOGAN (B.). – *Integrating BDI agents with agent-based simulation platforms*. Autonomous Agents and Multi-Agent Systems. vol. 30, no. 6, pp. 1050-1071 (2016).
- [59] SMITH (R.G.). – *The contract net protocol: High-level communication and control in a distributed problem solver*. IEEE Computer Architecture Letters. vol. 29, no. 12, pp. 1104-1113 (1980).
- [60] SWARUP (S.) et MORTVEIT (H.S.). – *Live Simulations*. In: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, p. 1721-1725 (2020).
- [61] TAILLANDIER (P.), BOURGAIS (M.), CAILLOU (P.) et al. – *A BDI agent architecture for the GAMA modeling and simulation platform*. In: International Workshop on Multi-Agent Systems and Agent-Based Simulation. Springer, p. 3-23 (2016).
- [62] TAILLANDIER (P.), GAUDOU (B.), GRIGNARD (A.) et al. – *Building, composing and experimenting complex spatial models with the GAMA platform*. Geoinformatica, vol. 23, no. 2, pp. 299-322 (2019).
- [63] TISUE (S.) et WILENSKY (U.). – *Netlogo: A simple environment for modeling complexity*. In: International conference on complex systems. Boston, MA, p. 16-21 (2004).
- [64] TREUIL (J.-P.), DROGOUL (A.) et ZUCKER (J.-D.). – *Modélisation et simulation à base d'agents: exemples commentés, outils informatiques et questions théoriques*. Dunod (2008).
- [65] TUYLS (K.) et WEISS (G.). – *Multiagent learning: Basics, challenges, and prospects*. AI Magazine, vol. 33, no. 3, pp. 41-41 (2012).
- [66] VÁZQUEZ-SALCEDA (J.), DIGNUM (V.) et DIGNUM (F.). – *Organizing multiagent systems*. Autonomous Agents and Multi-Agent Systems, vol. 11, no. 3, pp. 307-360 (2005).
- [67] VO (D.-A.), DROGOUL (A.) et ZUCKER (J.-D.). – *An operational meta-model for handling multiple scales in agent-based simulations*. In: 2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future. IEEE., p. 1-6 (2012).
- [68] WINIKOFF (M.) et PADGHAM (L.). – *The prometheus methodology*. In: Methodologies and software engineering for agent systems. Springer, p. 217-234 (2004).
- [69] WOOLDRIDGE (M.). – *An introduction to multiagent systems*. John Wiley & sons (2009).
- [70] WOOLDRIDGE (M.J.) et JENNINGS (N.R.). – *Intelligent agents: Theory and practice*. The knowledge engineering review, vol. 10, no. 2, pp. 115-152 (1995).
- [71] WOOLDRIDGE (M.), JENNINGS (N.R.) et KINNY (D.). – *The Gaia methodology for agent-oriented analysis and design*. Autonomous Agents and multi-agent systems, vol. 3, no. 3, pp. 285-312 (2000).

GAGNEZ DU TEMPS ET SÉCURISEZ VOS PROJETS EN UTILISANT UNE SOURCE ACTUALISÉE ET FIABLE

Techniques de l'Ingénieur propose la plus importante collection documentaire technique et scientifique en français !

Grâce à vos droits d'accès, retrouvez l'ensemble des **articles et fiches pratiques de votre offre**, **leurs compléments et mises à jour**, et bénéficiez des **services inclus**.



RÉDIGÉE ET VALIDÉE
PAR DES EXPERTS



MISE À JOUR
PERMANENTE



100 % COMPATIBLE
SUR TOUS SUPPORTS
NUMÉRIQUES



SERVICES INCLUS
DANS CHAQUE OFFRE

- + de 350 000 utilisateurs
- + de 10 000 articles de référence
- + de 80 offres
- 15 domaines d'expertise

- | | |
|---|---|
| <input type="radio"/> Automatique - Robotique | <input type="radio"/> Innovation |
| <input type="radio"/> Biomédical - Pharma | <input type="radio"/> Matériaux |
| <input type="radio"/> Construction et travaux publics | <input type="radio"/> Mécanique |
| <input type="radio"/> Électronique - Photonique | <input type="radio"/> Mesures - Analyses |
| <input type="radio"/> Énergies | <input type="radio"/> Procédés chimie - Bio - Agro |
| <input type="radio"/> Environnement - Sécurité | <input type="radio"/> Sciences fondamentales |
| <input type="radio"/> Génie industriel | <input type="radio"/> Technologies de l'information |
| <input type="radio"/> Ingénierie des transports | |

**Pour des offres toujours plus adaptées à votre métier,
découvrez les offres dédiées à votre secteur d'activité**

Depuis plus de 70 ans, Techniques de l'Ingénieur est la source d'informations de référence des bureaux d'études, de la R&D et de l'innovation.

www.techniques-ingenieur.fr

CONTACT : Tél. : + 33 (0)1 53 35 20 20 - Fax : +33 (0)1 53 26 79 18 - E-mail : infos.clients@teching.com

LES AVANTAGES ET SERVICES compris dans les offres Techniques de l'Ingénieur

ACCÈS



Accès illimité aux articles en HTML

Enrichis et mis à jour pendant toute la durée de la souscription



Téléchargement des articles au format PDF

Pour un usage en toute liberté



Consultation sur tous les supports numériques

Des contenus optimisés pour ordinateurs, tablettes et mobiles

SERVICES ET OUTILS PRATIQUES



Questions aux experts*

Les meilleurs experts techniques et scientifiques vous répondent



Articles Découverte

La possibilité de consulter des articles en dehors de votre offre



Dictionnaire technique multilingue

45 000 termes en français, anglais, espagnol et allemand



Archives

Technologies anciennes et versions antérieures des articles



Impression à la demande

Commandez les éditions papier de vos ressources documentaires



Alertes actualisations

Recevez par email toutes les nouveautés de vos ressources documentaires

*Questions aux experts est un service réservé aux entreprises, non proposé dans les offres écoles, universités ou pour tout autre organisme de formation.

ILS NOUS FONT CONFIANCE



www.techniques-ingenieur.fr

CONTACT : Tél. : + 33 (0)1 53 35 20 20 - Fax : +33 (0)1 53 26 79 18 - E-mail : infos.clients@teching.com