
Modélisation de systèmes complexes distribués : l'ingénierie des modèles pour l'intégration des paradigmes « agent » et « composant »

Philippe Aniorté¹, Eric Cariou² et Eric Gouardères²

¹ Laboratoire Informatique de l'Université de Pau et des Pays de l'Adour (LIUPPA)
IUT de Bayonne – Département Informatique
Place Paul Bert – 64100 Bayonne
aniorte@iutbayonne.univ-pau.fr

² Laboratoire Informatique de l'Université de Pau et des Pays de l'Adour (LIUPPA)
Université de Pau – Département Informatique – B.P. 1155
64013 Pau CEDEX
Eric.Gouarderes@univ-pau.fr, Eric.Cariou@univ-pau.fr

RÉSUMÉ. Cet article s'intéresse dans un premier temps à la modélisation et à la simulation des systèmes complexes qui constituent notre champ d'investigation. Après avoir mis en évidence les besoins de cette communauté, des éléments de solution basés sur le paradigme « agent » sont listés. Dans un second temps nous présentons les grandes lignes d'Ugatze, un modèle de composants logiciels développé au sein de l'équipe, dédié à l'ingénierie de systèmes complexes distribués, basé sur le concept d'interaction. Enfin, il décrit l'état de nos réflexions concernant le recours à l'ingénierie des modèles (IDM ou MDE) pour intégrer les paradigmes « agent » et « composant » et définir une architecture de haut niveau pour la simulation distribuée.

MOTS-CLÉS : MODELISATION, SIMULATION, AGENT, COMPOSANT, INGENIERIE DES MODELES (IDM, MDE)

Introduction

Jusqu'il y a peu, les communautés « composant » et « agent » étaient relativement cloisonnées, dans le sens où peu d'équipes travaillaient à l'intersection de ces deux paradigmes. Le domaine des composants, que nous restreignons dans cet article aux « composants logiciels », est le creuset de nombreuses propositions, en matière de modèles (« industriels » et « académiques »), de démarches, et d'outils, issus essentiellement de la communauté « Génie Logiciel » mais également de la communauté « Systèmes d'Information ».

Dans le même temps, les Systèmes Multi-Agents et les propositions associées, issus de la communauté « Intelligence Artificielle », ont prouvé leur bien-fondé dans différents champs d'application, en particulier les systèmes complexes distribués, et plus spécialement la simulation. Récemment, une réflexion menée par des chercheurs issus des communautés citées précédemment a conduit à mettre en évidence des besoins croisés, et à s'interroger, entre autres, sur :

- des applications utilisant conjointement les deux approches,
- des méthodes d'analyse et de conception intégrant ces deux approches.

Le projet présenté dans cet article s'inscrit dans cette tendance. Il s'agit d'intégrer les résultats obtenus dans le domaine de la modélisation de structures d'organisations multi-agents, et ceux obtenus en matière de modélisation de composants et d'interactions (entre composants). Plus précisément, l'objectif consiste, en s'appuyant sur les principes de l'Ingénierie Dirigée par les Modèles (IDM ou MDE), à élaborer des propositions visant à construire un modèle d'architecture de haut niveau pour la simulation distribuée permettant l'intégration de composants de simulation, de systèmes opérationnels et d'opérateurs humains (superviseurs, décideurs, utilisateurs). Outre ce modèle d'architecture, nous envisageons des outils d'édition et de transformation de modèles pour permettre la projection vers des plates-formes de mise en œuvre. En particulier, des outils formels devront garantir la validité des transformations. Le champ d'application de ces travaux est la formation aéronautique (pilotage, maintenance...).

Dans un premier temps, nous présentons la problématique de la communauté « Modélisation et Simulation » (M&S) ainsi que les avantages du paradigme « agent » en la matière. La deuxième partie vise à présenter les grandes lignes d'un modèle de composants logiciels autonomes conçu au sein de l'équipe et basé sur le concept d'interaction. Puis, nous décrivons dans la troisième partie l'état de nos réflexions concernant l'apport de l'approche IDM, support à l'intégration des paradigmes « agent » et « composant » pour la conception de systèmes de simulation distribués.

1. Modélisation et simulation des systèmes complexes

La communauté Modélisation et Simulation (M&S) est née des besoins induits par l'étude et la modélisation des systèmes complexes mais aussi par leur exploitation et leur maintenance. Un système complexe est défini comme un ensemble d'éléments en interaction entre eux et avec l'extérieur. Par essence il s'agit de systèmes ouverts, hétérogènes dans lesquels les interactions sont non linéaires et dont le comportement global ne peut être obtenu par simple composition des comportements individuels.

1.1. Problématique

Depuis une dizaine d'années, de nouvelles architectures de systèmes d'information ont émergé. Elles sont caractérisées par l'absence d'autorité centrale, des composants faiblement couplés et la capacité à répondre – réactivité – et à s'adapter – flexibilité – rapidement aux nouveaux besoins et aux nouvelles contraintes selon des principes d'auto-organisation [MASS99]. Le World Wide Web en est l'exemple le plus connu, mais on peut trouver des exemples dans le domaine de la production (chaînes logistiques, e-manufacturing), des transactions financières (e-business) de la formation (e-learning) et l'entraînement (e-training). Par contre ces nouvelles architectures ont eu peu d'impact dans le domaine de la simulation jusqu'au milieu des années 90 où le concept de "Simulation basée Web" (Web Based Simulation, WEBSIM <http://www.websim.net/>) a été proposé, et est notamment supporté depuis par l'OMG. L'objectif est d'encourager le développement et l'application de standards ouverts et de démarches conceptuelles qui permettent à la communauté M&S d'exploiter ces nouvelles architectures afin d'intégrer leurs produits dans des systèmes à plus grande échelle. La vision est celle d'un système distribué basé sur une intégration horizontale de composants M&S, de systèmes opérationnels et d'opérateurs humains (superviseurs, décideurs, utilisateurs). Chaque partie de ce système étant autonome, interfaçable avec les autres parties pour fournir ou consommer des données en temps réel.

Parallèlement, le Département de la Défense américain (DoD) a investi de façon significative pour soutenir des projets dans le domaine de la simulation distribuée. Il a encouragé l'évolution de standards pour supporter la réutilisation et l'interopérabilité de simulations ce qui a abouti à HLA (High Level Architecture [DAH97]) standardisé par l'OMG (HLA 1.3) et l'IEEE (HLA 1516) et peut être reconnu aujourd'hui comme un standard de facto pour résoudre les problèmes liés à l'intégration, la réutilisation et l'interopérabilité dans le domaine de la simulation distribuée. Si c'est effectivement le cas dans le domaine militaire, la diffusion dans l'industrie non militaire des technologies de simulation distribuée reste confidentielle. Les raisons sont multiples et analysées dans un article de synthèse par un ensemble de chercheurs et de professionnels du domaine [TAYL02]. On peut citer :

4 JMAC 2006

- Le manque de méthodologies et d'outils pour aborder la modélisation d'un système complexe à différents niveaux de granularité.
- Le manque de standards pour décrire les interactions (échanges de données) entre composants.
- Le manque de flexibilité : capacité d'adaptation et de configuration dynamique du système. La flexibilité doit permettre de simuler des scénarios complexes fondés sur les points de vue multiples de chaque acteur (simulateur, opérateur humain) où les règles, les contrats et la structure du réseau peuvent changer en cours de simulation.

1.2. Un élément de solution : le paradigme « agent »

Les systèmes multi-agents peuvent apporter des contributions significatives aux limites énoncées ci-dessus en fournissant des modèles et des méthodologies de conception de haut niveau basés sur une approche centrée organisation (OCMAS, Organization Centered Multi-Agent Systems, [FERB03]), des protocoles standards d'interaction adaptés (ex : FIPA Interaction protocols¹) et des plates-formes d'exécution distribuées (ex: MADKIT², JADE³). En terme d'architecture de simulation distribuée, une approche consiste à construire des applications de simulation multi-agents au-dessus de HLA. ACES [ARON03] est un exemple d'architecture flexible et modulaire exploitant le paradigme agent et HLA, pour l'analyse des systèmes d'échange dans l'aviation civile. Actuellement, le projet Federation Grid⁴ a pour objectif le développement d'un environnement de simulation flexible proposé comme un ensemble de web (grille) services pour les jeux de stratégies. Dans ces deux exemples, HLA fournit un support pour l'interopérabilité des différents composants et les agents sont utilisés pour la flexibilité et la modularité du système global. C'est pourquoi un des enjeux de ce qui est appelé aujourd'hui « Agent Directed Simulation » est de fournir des services d'agents « plug and play » pour les applications futures sur la grille.

Agent-Directed Simulation [OREN00] est proposé comme un nouveau paradigme qui couvre les différentes approches et relations entre agents et simulation. Il s'agit de promouvoir, au sein de la communauté modélisation et simulation, l'utilisation des agents logiciels comme assistants pour le développement de systèmes et la conduite d'expérimentations en simulation. Trois types de synergies ont été identifiées: (1) au niveau conceptuel, les agents sont utilisés comme métaphores pour la modélisation de systèmes (agent simulation), (2) au niveau développement, les agents sont utilisés comme métaphore de programmation pour faciliter la réutilisation de modèles et de services de simulation (agent supported

¹ FIPA, Foundation for Intelligent Physical Agents, FIPA Interaction Protocols Technical Committee, http://www.fipa.org/activities/interaction_protocols.html, 2003

² Multi Agent Development KIT. <http://www.madkit.org>

³ <http://sharon.csel.it/projects/jade/>

⁴ <http://www.federationgrid.org/>, <http://www.magnetar.org/sdk/>

simulation), (3) les agents utilisent la simulation comme méthode de décision ou la simulation est utilisée comme technique expérimentale pour évaluer le comportement de sociétés d'agents (agent based simulation).

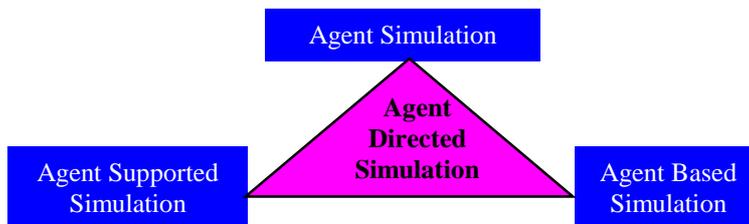


Figure 1 : Agent Directed Simulation

Nous avons exploité en particulier les synergies mises en évidence ci-dessus, pour proposer un cadre de développement d'environnement de simulation de systèmes de pilotage de la production [GOUA05]. La contribution principale est la modélisation de structures de contrôles distribuées. La structure organisationnelle régissant les interactions s'appuie sur le modèle AALAADIN [GUTK99], aujourd'hui AGR (Agent Groupe Rôle). Les groupes sont créés pour prendre en compte des objectifs (respect des délais, des quantités, minimisation des coûts...). Une structure abstraite précise les relations structurelles entre rôles au sein des groupes, ce qui permet une description simple de structures d'organisations et d'interactions quel que soit le niveau de granularité considéré (par exemple cellule, Atelier, Entreprise) [TCHI04].

Nous avons choisi AGR car c'est un méta-modèle qui respecte notamment trois principes le rendant générique pour décrire différents modèles organisationnels [FERB03]: (1) le niveau organisationnel décrit le "quoi" et non le "comment"; Il borne le domaine d'activité des agents mais ne décrit pas le comportement de ceux-ci, (2) aucune contrainte ne doit être posée sur la dimension cognitive des agents, qui peuvent être intentionnels ou réactifs et (3) une organisation est un moyen de partitionner un système. Chaque partition ou groupe définit le contexte d'interaction des agents de du groupe. En particulier des agents de groupes différents ne peuvent pas communiquer directement. D'autres modèles organisationnels tels que STEAM [TAMB97] ou MOISE [HANN00] auraient pu être envisagés mais ils sont plus spécifiques et concernent des agents intentionnels (notion d'engagement à partir de buts et de plans pour les réaliser). Nous avons privilégié plutôt une approche émergente.

2. Un modèle de composants autonomes

Au cours de ces dernières années, l'équipe a développé et expérimenté (dans le cadre du projet européen ASIMIL) un (méta)modèle de composants dénommé « Ugatze », adapté à la réutilisation de composants logiciels autonomes, hétérogènes

et distribués. Ce modèle de composants est défini précisément via un méta-modèle, ce qui permet de manipuler des modélisations Ugate via des outils dans le cadre d'un processus logiciel de type « Ingénierie Dirigée par les Modèles » (IDM ou MDE pour Model-Driven Engineering – voir section 3). Ces aspects n'étant pas développés dans cet article, nous renvoyons le lecteur intéressé à [ANIO04]. Ce méta-modèle repose sur deux notions essentielles :

- l'interface du composant. C'est le résultat de la (re)spécification du composant, activité propre à la réutilisation [CAUV99]. La représentation de tous les composants à réutiliser est « unifiée » (au sens des besoins exprimés pour l'interopérabilité) via le méta-modèle.
- l'interaction entre composants. Elle permet de gérer l'interopérabilité entre composants.

2.1. L'interface du composant

La spécification des composants réutilisables telle que nous la pratiquons repose sur le concept « d'interface » constituée de « points d'interaction ». Ils sont dédiés à l'intégration du composant dans le système à développer, rendant effective sa réutilisation.

Nous distinguons différents types de points d'interaction. L'autonomie des composants et le principe de découplage associé, reposant entre autres sur la séparation du flot d'information et du flot de contrôle [LHUI98], nous conduisent à recenser :

- des points d'interaction de donnée,
- des points d'interaction de contrôle.

Considérant l'aspect flux, nous distinguons :

- des points d'interaction d'entrée, flux entrant dans le composant,
- des points d'interaction de sortie, flux sortant du composant.

En croisant les deux critères, donnée / contrôle et entrée / sortie, nous obtenons :

- des points d'interaction d'entrée de donnée ou Data Input (DI),
- des points d'interaction de sortie de donnée ou Data Output (DO),
- des points d'interaction d'entrée de contrôle ou Control Input (CI),
- des points d'interaction de sortie de contrôle ou Control Output (CO).

2.2. L'interaction entre composants

Le méta-modèle proposé est basé sur le principe de découplage entre composants. L'interaction s'appuie sur ce principe. Elle permet de gérer l'interopérabilité conceptuelle entre composants.

Les composants réutilisables, modélisés à l'aide des points d'interaction, sont destinés à être intégrés afin de développer le nouveau système. Au sein de ce

système, ils doivent pouvoir interopérer. L'interopérabilité est gérée à l'aide du concept d'interaction. La mise en œuvre des interactions est basée sur les points d'interaction présentés précédemment. Nous distinguons :

- interactions « prédéfinies ». Elles constituent pour le concepteur « l'offre de base » en matière d'interactions. Cette offre se caractérise par des dispositifs généraux, dont le nombre est limité, utilisés de façon récurrente. A l'instar des points d'interaction, les interactions prédéfinies se déclinent en deux catégories : interactions d'information et interactions de contrôle.

- interactions « à façon ». Elles constituent pour le concepteur une alternative à l'offre de base. L'idée est de donner au concepteur la possibilité de définir une interaction adéquate, comme solution à un problème spécifique.

De façon théorique, ce dernier type d'interactions offre une solution au problème classique du nombre fini de dispositifs, nécessairement non suffisant pour traiter la variété des problèmes pouvant se poser. De ce point de vue, l'analogie avec les types dans les langages de programmation nous semble pertinente. La seule limite tient à l'expressivité de la forme algorithmique, car le concepteur est invité à définir ces interactions à l'aide d'algorithmes. Plus concrètement, cette possibilité contribue grandement à la gestion de composants autonomes, en autorisant un haut niveau de découplage, ce qui dans le contexte de nos travaux est fondamental. En conséquence, la prise en compte de l'hétérogénéité s'en trouve facilitée. In fine, la gestion de l'interopérabilité est favorisée.

2.3. Représentation graphique et règles

Outre les notions clés déjà présentées (interface et interaction), le méta-modèle « Ugate » se caractérise par :

- une représentation graphique. Elle permet d'instrumenter son utilisation en offrant la possibilité de manipuler les points de spécification de l'interface des composants et les interactions entre composants basées sur ces points.

- des règles de bonne utilisation. Elles permettent de vérifier les modèles instanciés à partir du méta-modèle.

Nous ne détaillons pas la représentation graphique des interactions. Nous renvoyons le lecteur intéressé à [SEYL04]. Cette représentation graphique est complétée par des règles. De telles règles ont pour objet de spécifier un ensemble de contraintes relatives aux différents éléments constitutifs du méta-modèle, interface (points de spécification) et interactions. Elles sont destinées à aider à sa bonne utilisation.

Ces règles correspondent typiquement aux règles de bonne construction (« well-formedness rules ») que l'on retrouve dans la définition de méta-modèles. Elles sont décrites en langage naturel et à l'aide du langage OCL (Object Constraint Language).

3. L'ingénierie des modèles appliquée à la simulation distribuée

L'ingénierie des modèles – ou MDE (pour Model-Driven Engineering) – est une discipline émergente qui vise à définir un nouveau mode de développement des logiciels [AS-MDA04]. Son but est de déplacer la complexité de réalisation des applications logicielles de l'écriture du code du logiciel (la programmation) vers la spécification de l'application (la modélisation). L'objectif principal du MDE est de se baser sur la spécification abstraite (une modélisation indépendante de toute mise en œuvre, de toute technologie) d'une application comme support pour générer automatiquement le code de cette application pour une plate-forme technologique donnée. Pour passer de cette spécification abstraite au code, les modèles sont transformés via une série de transformations correspondant à des raffinements ou des projections afin d'aller vers une spécification détaillant l'implémentation et enfin générer le code. Le MDE est donc une nouvelle « philosophie » de développement où les modèles sont de plus en plus considérés avec une vision productive (dans le sens où l'implémentation sera directement dérivée des modèles de haut niveau) au lieu de contemplative (dans le sens où le développeur consulte les modèles pour écrire son code).

Les technologies et thématiques de recherche autour du MDE concernent un nombre important de points comme, entre autres, des techniques et outils de définition et de gestion de méta-modèles (afin de pouvoir manipuler formellement des modèles) ainsi que des techniques et outils de transformation de modèles (pour réaliser des projections, des raffinements ou bien encore des fusions de modèles). Ces technologies sont centrales dans le cadre de notre projet et elle seront utilisées pour intégrer les approches composants et agents dans le contexte de la simulation distribuée.

Les travaux à mener pour aboutir à cette intégration peuvent être classés en deux catégories : modélisation et transformation. En ce qui concerne la partie modélisation, il s'agit de définir un ensemble de méta-modèles qui permettront de manipuler et gérer tous les types de modèles requis. Cela impliquera notamment de définir (ou spécifier plus formellement) les méta-modèles d'organisation multi-agents (comme le méta-modèle AGR) et les méta-modèles de simulation (comme HLA), puis en déduire un méta-modèle abstrait commun. A ces méta-modèles s'ajoute aussi le méta-modèle de composants Ugatez déjà existant. Pour la partie transformation, il s'agira de définir un ensemble de transformations permettant de projeter un modèle d'agents respectant notre méta-modèle abstrait vers un modèle technologique de mise en œuvre comme HLA ou bien encore le modèle de composants Ugatez. Les transformations seront de deux types : (1) transformation d'une spécification abstraite en une spécification concrète par rapport à une plate-forme technologique et (2) génération de code pour une plate-forme donnée à partir d'une spécification concrète. Dans ce cadre de transformation, une attention particulière sera portée à la définition d'outils permettant de valider les transformations de modèles [CMSD04]. Les relations entre les modèles, méta-

modèles et les diverses transformations sont représentées sur la figure 2. Une application est spécifiée à un niveau abstrait, via les concepts de notre méta-modèle abstrait. Ensuite, cette spécification abstraite est par exemple projetée en une spécification concrète par rapport à la plate-forme HLA. Enfin, à partir de cette spécification concrète, le squelette du code de l'application implémentée sur HLA est généré.

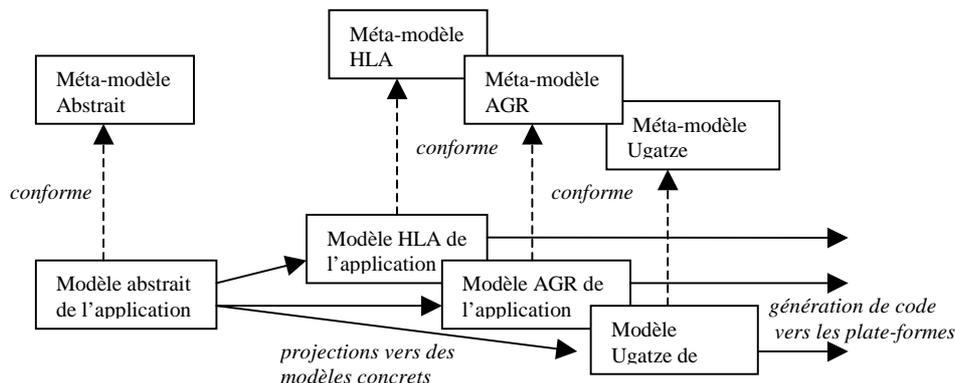


Figure 2 : Modèles et méta-modèles

Notre objectif est donc d'être capable tout d'abord de spécifier une application de simulation distribuée à un niveau abstrait, indépendamment de la plate-forme de mise en œuvre. Ensuite, des transformations de modèles permettront de projeter une telle spécification vers plusieurs plates-formes, soit de type multi-agents, soit de type composants soit de type dédiée comme HLA. Il est également possible d'envisager des projections « mixtes » comme par exemple une mise en œuvre utilisant des composants Ugatze par-dessus une plate-forme multi-agents ou bien encore la mise en œuvre d'une structure d'organisation de type AGR au dessus d'une plate-forme HLA.

La définition du méta-modèle abstrait et des concepts qu'il intègre est évidemment un point crucial. Deux approches sont possibles. La première consiste à prendre le plus grand ensemble des concepts communs ou proches à tous les méta-modèles concrets considérés (on pourra notamment y trouver un concept mixte entre agent et composant qui correspondrait directement au concept de composant dans une plate-forme composants et au concept d'agent dans un approche multi-agents). La seconde approche consiste à définir un ensemble plus large de concepts intégrant des concepts qui ne sont pas présents dans tous les méta-modèles. Par exemple, on intégrerait les notions de groupe et de rôles qui ne sont pas transposables directement en concepts d'une plate-forme composants. Il faudrait alors définir une règle de projection traduisant un concept abstrait en un ensemble d'éléments concrets pour une plate-forme donnée (au lieu d'une correspondance directe entre un concept abstrait et un concept concret comme pour la première approche). Cette seconde approche est évidemment plus complexe à mettre en œuvre, mais elle

présente l'avantage de définir un méta-modèle plus fourni et donc des modèles abstraits plus évolués.

4. Conclusion

Cet article est une contribution à porter au crédit des travaux se situant à l'intersection des paradigmes composant et agent. D'une part, il concerne des applications pouvant bénéficier conjointement des deux approches. En l'occurrence, il s'agit des nombreuses applications issues du domaine de la simulation distribuée, dont nous avons dressé les besoins au cours de la première partie.

D'autre part, cet article concerne aussi les méthodes d'analyse et de conception intégrant ces deux approches. Plus précisément, il s'agit d'intégrer en la matière les avantages du paradigme « composant » et du paradigme « agent ». Le premier a démontré ses atouts pour la réutilisation et la modélisation. Le modèle Ugatez développé au sein de l'équipe et ébauché dans la deuxième partie en est un exemple. Par ailleurs, le paradigme « agent » a donné lieu à un certain nombre de propositions qui constituent des éléments de réponse aux problèmes de la communauté M&S.

Notre contribution vise à recourir à l'approche MDE, basée entre autre sur la méta-modélisation et les transformations de modèles, pour intégrer les avantages de ces deux paradigmes, afin de définir des architectures de haut niveau pour la simulation distribuée. Nos propositions, présentées dans la troisième partie, nécessitent cependant d'être approfondies, ce qui constitue la perspective majeure des travaux présentés dans cet article.

Bibliographie

- [ANIO04] Anioté P. - Vers des systèmes distribués et hétérogènes : Une approche basée composants guidée par les modèles - Habilitation à Diriger des Recherches, Université de Pau et des Pays de l'Adour, 21 octobre 2004
- [ARON03] Aronson J., Manikonda V., Peng W., Levy R., and Roth K. 2003. "An hla compliant agent-based fast-time simulation architecture for analysis of civil aviation concepts". In proc. of Spring SISO Simulation Interoperability Workshop, Orlando.
- [AS-MDA04] Rapport de Synthèse de l'Action Spécifique (AS) CNRS sur l'Ingénierie Dirigée par les Modèles (IDM), Jean Bézivin, Mireille Blay, Mokrane Bouzeghoub, Jacky Estublier, Jean Marie Favre, Sébastien Gérard, Jean-Marc-Jézéquel - 2004, <http://www-adele.imag.fr/mda/as/>
- [CAUV99] Cauvet C., Semmak F. - La réutilisation dans l'ingénierie des systèmes d'information - Dans Génie objet - Analyse et conception de l'évolution, Sous la direction de Oussalah C., pp. 25-55, Hermès, 1999
- [CMSD04] Eric Cariou, Raphaël Marvie, Lionel Seinturier, and Laurence Duchien. OCL for the Specification of Model Transformation Contracts. Workshop OCL and Model Driven

Engineering of the Seventh International Conference on UML Modeling Languages and Applications (UML 2004), October 2004.

- [DAH97] Dahmann, J.S., R.M. Fujimoto, and R.M. Weatherly. . The Department of Defense High Level Architecture. In Proceedings of the 1997 Winter Simulation Conference, ed. S. Andradóttir, K.J. Healy, D.H. Withers, and B.L. Nelson, 142-149, Association of Computing Machinery, New York, NY, 1997.
- [FER03] Ferber J., Gutknecht O., Michel F. 2003. From Agents to Organizations: an Organizational View of Multi-Agent Systems. In proceedings of Fourth International Workshop on Agent Oriented Software Engineering (AOSE), 2003.
- [GOU05] Gouardères E., Tchikou M., Lamarque N., Agent-based framework for simulation in manufacturing control. Proceedings of the 2005 European Simulation and Modelling Conference, EUROSIS-ETI Publication, pp 95-100, Porto, Portugal, 2005.
- [GUT99] Gutknecht, O. , Ferber, J. (1999). Vers une méthodologie organisationnelle de conception de systèmes multi-agents. Rapport de recherche de LIRMM, n°99073, 1999.
- [HAN00] Hannoun M., Boissier O., Sichman J., Sayettat C. MOISE: An Organizational Model for Multi-agent Systems. Monard, Sichman Eds, IBERAMIA/SBIA, Springer LNAI 1952, 2000.
- [LHU98] Lhuillier M. - Une approche à base de composants logiciels pour la conception d'agents - Principes et mise en œuvre à travers la plateforme MALEVA, Thèse de l'Université Paris VI, Février 1998
- [MAS99] Massote P. Auto-organisation dans les structures et les systèmes, Actes de la deuxième conférence francophone de Modélisation et SIMulation, p. 21-29, 1999.
- [ORE00] Ören T.I., Numrich S.K., Wilson L.F., Uhrmacher A. M., Gelenbe E. "Agent-directed simulation - challenges to meet defense and civilian requirements". In Proc. of the 2000 Winter Simulation Conference. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds., 2000.
- [SEY04] Seyler F. - Ugatze : métamodélisation pour la réutilisation de composants hétérogènes distribués - Thèse de l'Université de Pau et des Pays de l'Adour, 16 décembre 2004.
- [TAM97] Tambe M. Towards flexible teamwork. Journal of Artificial Intelligence research (JAIR), n°7, pp. 83-124, 1997.
- [TAY02] Taylor S. J E., Fujimoto R., Straßburger S., Bruzzone A., Boon P. G., Ray J.P. Distributed Simulation and Industry : Potentials and pitfalls. In Proc. of the 2002 Winter Simulation Conference, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds., 2002.
- [TCH04] Tchikou M., Pilotage temps réel basé sur les concepts d'organisation et d'interaction SMA . Xèmes Journées Francophones sur les Systèmes Multi-Agents, Hermès Science publications, ISBN 2-7462-1021-5, pp. 209-222, Paris, France, 24-26 Novembre 2004.