A Comparative Analysis of Multiagent System Development Methodologies: Towards a Unified Approach

Arsène Sabas
Département d'informatique,
Université Laval,
Québec, Canada, G1K 7P4
email: sabas@iad.ift.ulaval.ca

Sylvain Delisle & Mourad Badri
Département de mathématiques et d'informatique,
Université du Québec à Trois-Rivières,
Québec, Canada, G9A 5H7
email: {Sylvain_Delisle, Mourad_Badri}@uqtr.ca

Abstract

Agent-oriented methodologies and related modeling techniques have become a priority for the development of large scale agent-based systems. Several methodologies have been proposed for the development of multiagent systems (MAS). For the most part, these methodologies remain incomplete: they are either an extension of object-oriented methodologies or an extension of knowledge-based methodologies. In addition, too little effort has gone into the standardization of MAS methodologies, platforms and environments. We present a comparative analysis of the main existing MAS methodologies. Our long-term goal is the development of a rigorous and complete methodology for the analysis and design of MAS.

1 Introduction

The work we present here belongs to the disciplines of Software Engineering and Distributed Artificial Intelligence. The increased interest in using MAS in all kinds of computer system applications has created a need for stronger foundations in MAS software engineering. Several methodologies have been proposed for the development of MAS, but they remain incomplete: they are either an extension of object-oriented methodologies (e.g. Agent Modeling for System of BDI agents [Kinny et al, 1996]) or an extension of knowledge-based methodologies (e.g. CoMoMAS [Glaser, 1996]). These various methodologies do not provide adequate specifics to model agents' characteristics, such as their mental state or their social behaviour in the context of MAS. Even methodologies especially developed for MAS, such as GAIA [Wooldridge et al, 2000] do not truly take into account some fundamental aspects like validation.

It is our conviction that too little effort has gone into the standardization of MAS methodologies, platforms and environments. The success of the agent paradigm requires systematic methodologies for the specification, analysis and design of "non toy" MAS applications. The work we discuss here constitutes a step towards this goal. We present a comparative analysis of existing MAS methodologies which could eventually lead to a general specification for the design of a complete MAS methodology. To support our comparison task, we have designed a new grid called the MUltidimensional framework of Criteria for the Comparison of MAS methodologies (MUCCMAS).

2 The MUCCMAS framework

Pascot & Bernadas (1993) have proposed in 1993 a framework for the comparison of design methods for computerized information systems. Their framework was adapted by [Adam and Kolki, 1999] to the analysis and design of interactive systems dedicated to complex administrative systems. The criteria we selected in our MUCCMAS framework were inspired by those of [Adam & Kolki, 1999], as well as those defined for the comparative analysis of MAS platforms in [AFIA/PRC, 1998]. However, we have customized these criteria to the domain of MAS development. Another important source of information for us during this research was the work of [Flores-Mendez, 1999]. One element of originality in our framework is the use and adaptation of concepts from object-oriented software engineering to the development of MAS methodologies. MUCCMAS is comprised of six dimensions. The authors of [Adam & Kolki, 1999] added the COOPERATION dimension to the four dimensions (METHODOLOGY, REPRESENTATION, ORGANISATION, and TECHNOLOGY) of [Pascot & Bernadas, 1993]. We have added to these five dimensions a sixth called the **AGENT dimension**.

2.1 The Methodology Dimension

The METHODOLOGY dimension contains 7 criteria:

- stages of the process (various phases that exist in the development process of the methodology: analysis, modeling, specification, design, validation, verification and ergonomic evaluation, etc.);
- development models (cascade, V, spiral, incremental, nabla [Kolski, 1997] , etc.);
- development approach (top-down, bottom-up, evolving, etc.);

- degree of user implication (indicates whether the methodology provides communication means between designers and users and, also, how deeply the user is involved during the development);
- moment of user implication (beginning, middle or end of project);
- models re-use (indicates whether the methodology provides or makes it possible to provide a repository of reusable models);
- availability of software or methodological support (indicates whether there are available tools to support the methodology).

2.2 The Representation Dimension

Once system requirements have been completed, the next step is the elaboration of a model that will translate the system's outline from an external point of view to a more precise and rigorous description. The latter, as prepared by system engineers and architects, is usually a representation amenable to graphical depictions that will facilitate comprehension and communication between experts on the one hand, and expert and users on the other hand. The REPRESENTATION dimension contains 4 criteria:

- system division (levels of abstractions, generalization-specialization, type-occurrence, strategy-tactic, etc.);
- formalism (diagrams, concepts and rules used in the methodology);
- sequencing (derivation and the relationship between various models of the methodology);
- quality of the models used in the methodology (number of models, cohesion of the models, coverage of the models, complexity of the models, etc.).

2.3 The Agent Dimension

This dimension allows the description of agents' characteristics in the specific MAS to which the development methodology is applied. System performance, efficacy and efficiency are related to these characteristics. Features of the agents constitute a determining factor in their social and cooperative behaviour. Agent-based system designers would like to endow the agents with an anticipation and planning capacity that would allow them to optimize their (individual and collective) behaviour. Four criteria are associated with the AGENT dimension:

- nature of the agents (homogeneous or heterogeneous);
- types of agents (intelligent agents, interfaces or personal agents, mobile agents, information agents, autonomous agents, etc.);
- agent attributes (intrinsic agent characteristics that the methodology uses: adaptability, autonomy, cooperative behaviour, inferential capability, "knowledge-level" communication ability, mobility, personality, reactivity, temporal continuity, deliberative behaviour, etc.);
- agent features (how to predict the behaviour of an agent, not knowing its internal structure? Which representations associated with an agent best define its observable or expected behaviour. Dennett's [Dennett, 1987] three stances attempt to cover these types of representations: physical stance, design stance, intentional stance).

In order to manage the dynamics of the organizations in the MAS, it is necessary to endow each agent with a module of recognition or intention attribution.

2.4 The Organization Dimension

The structure of an organization is related to the environment in which it evolves, to the resources available to produce the outputs, as well as to the nature of these outputs. MAS development requires the construction of an organizational structure in order to be able to control the complexity of the system. This dimension defines the structure the system must have and the environment's characteristics for which the system is intended, and it also indicates whether the methodology explicitly specifies this structure and these characteristics. The four criteria associated with the ORGANIZATION dimension are:

- image of organization (hierarchical system, distributed system, open system, holonic system, etc.);
- nature of environment (structured, stable, explicit, determinist, observable, etc.);
- type of environment (active, passive);
- characteristics of the data processed (numerical, symbolic).

2.5 The Cooperation Dimension

This dimension is another significant aspect of MAS by which agents must cooperate to achieve a common goal. Many researchers have showed that cooperation between agents accelerates the problem resolution process and improves the results. In a methodology, it seems highly desirable to find applicable generic principles of cooperation, expressed in a systematic way, for all different kinds of systems the methodology is applicable to. For example, these principles should make it possible to establish and to maintain the cooperative state of the agents contained in the system. The six criteria of the COOPERATION dimension are:

- possible types of communication (communication between heterogeneous agents, agent-human communication);
- communication mode (direct, indirect, synchronous, asynchronous);
- communication language (based on signals, speech acts or others);
- cooperation model (identifies cooperation concepts used in interaction models of the methodology: negotiation, delegation of tasks, planning, etc.);
- type of control (centralized, hierarchical or distributed, etc.);
- interaction (static, dynamic, interaction engine distributed or centralized, interaction protocols explicit or implicit, interaction mechanisms to solve the non cooperative states between agents, etc.).

2.6 The Technology Dimension

The purpose of this dimension is to describe the characteristics of the software to which the methodology is applied. These characteristics constitute, in our opinion, an important parameter in the choice of a suitable methodology for a given application. The TECHNOLOGY dimension contains 5 criteria:

- mode of processing (batch, interactive, client-server, synchronous, asynchronous, distributed, etc.);
- human-machine interface type (classic, adaptable, adaptive, assistant, etc.);
- programming type (structured, object-oriented, agent-oriented, etc.);
- application type (simulation, problem resolution, integration, etc.).
- development environment (describes the development characteristics of the MAS to which the method can be applied (possible platforms, programming languages, other tools being used to implement the agents, etc).

3 Applying MUCCMAS

We have analyzed and compared nine well-known MAS development methodologies according to every dimension of our MUCCMAS framework:

1. A Methodology and Modeling Technique for Systems of BDI Agents (MMTS) [Kinny et al, 1996] ;
2. Agent-Oriented Design of Soccer Robot Team (Cassiopeia) [Collinot & Drogoul, 1996];
3. Agent-Oriented Methodology for Enterprise modeling (AOMEM) [Kendall et al, 1996] ;
4. An Agent-Oriented Methodology: High-Level and Intermediate Models (HLIM) [Ealmmari & Lalonde, 1999];
5. Analysis and Design of Multiagent Systems Using MAS-CommonKADS [Iglesias et al,1997] ;
6. GAIA [Wooldridge et al, 2000] ;
7. Multi-Agent Scenario-Based Method (MASB) [Moulin et al, 1996] ;
8. Multiagent Systems Engineering (MaSE) [Deloach, 1999] ;
9. The CoMoMAS Methodology and Environment for Multi-Agent System Development [Glaser, 1996].

We consider these 9 methodologies as representative of the main existing approaches. We could also have studied methods such as MESSAGE/UML [Giovanni et al, 2001]. MESSAGE/UML tries to combine the approaches of GAIA and MAS-CommonKADS by defining in a clear way the concepts required for agent-oriented analysis and a common semantic. The MESSAGE modeling language extends UML with agent-related concepts. We next present a summary of this comparative analysis.

3.1 The Methodology Dimension

Except for validation, verification and ergonomic evaluation, all the methodologies cover the other development phases even if the latter are not explicitly identified in methodologies like MMTS, CoMoMAS, AOMEN and Cassiopeia. MAS-CommonKADS implicitly takes into account verification and ergonomic evaluation. None of the nine methodologies clearly indicates the models that can support it, nor the approach that it uses. Except for MASB, the user's point of view seems rather remote in the other methodologies. All the methods (try to) provide reusable models and offer software or methodological supports. In particular, MAS-CommonKADS provides relatively elaborated support which, in our opinion, is a valuable plus for this methodology.

3.2 The Representation Dimension

All the methodologies have means to control the complexity of the system under development. Aspects of strategy and tactics of these systems were taken into account only by MAS-CommonKADS. The representation formalisms used for data, processing and activities, and system dynamics constitute an extension of object-oriented techniques or an extension of techniques from knowledge-based methods. We find the representation techniques used by the HLIM methodology (UCMs) particularly interesting as they can truly facilitate comprehension and communication between the various actors involved during software development. The formalism used by the MaSE methodology allows for automatic components reuse and facilitates verification. Derivation is clearly supported by the GAIA, MaSE, HLIM, CoMoMAS methodologies; it is implicitly supported by the AOMEN and Cassiopeia methodologies and remains rather fuzzy in MMTS, MASB and MAS-CommonKADS. The models of these methodologies do not seem to cover all dimensions of MAS development and thus remain incomplete. However, the models of the MAS-CommonKADS methodology appear almost complete. Models of the MAS-CommonKADS, MASB, MMTS and CoMoMAS methodologies are particularly complex, thus making more difficult their application in practice.

3.3 The Agent Dimension (see Table 1)

All methodologies considered here allow for the design of MAS based on heterogeneous agents. These agents can be of any type and have any attribute mentioned in this dimension (except for mobility, temporal continuity and deliberative behaviour). MAS-CommonKADS through its expertise model takes into account the deliberative behaviour of the agents. All these methodologies can represent features of the agents according to the design stance and the intentional stance. Only MASB seems to take into account the physical stance. MAS-CommonKADS, MMTS and MASB seem to have taken into account almost all the criteria of this AGENT dimension.

3.4 The Organization Dimension

In theory, all nine methodologies are able to deal with distributed systems. None of them can truly deal with open systems or open MAS. The organisational structures are not clearly identified in the methodologies, except for Cassiopeia in which it is a key concept. Nevertheless, the environment's nature is more or less taken into account by all the methodologies.

3.5 The Cooperation Dimension

All the methodologies can model communication between heterogeneous agents, and agents and humans. Except for GAIA, the communication mode used by these methodologies can be direct, synchronous and asynchronous. The communication language can be based on signals and speech acts, except for GAIA where this is not specified and for Cassiopeia which uses signals only. The cooperation models of these methodologies, except for GAIA, use the same concepts. The type of control, which is distributed in theory, remains implicit in these methodologies except for HLIM and MAS-CommonKADS where it is more or less clearly specified. Interaction is static in these methodologies and dynamic in methodologies MMTS, HLIM, MASB, MAS-CommonKADS, especially in Cassiopeia. All these methodologies account for interaction protocols.

Resolution of non cooperative states is not fully taken into account in these methodologies, except for GAIA, HLIM, CoMoMAS, MAS-CommonKADS and Cassiopeia where an interaction mechanism can solve simple conflicts. We found that cooperation receives little support in the GAIA methodology.

3.6 The Technology Dimension

All the methodologies are intended for problem resolution. The system implementation phase of these methodologies can use a structured or object-oriented programming approach. But we cannot clearly establish whether this programming can be considered agent-oriented since intrinsic characteristics of agent-oriented programming are not yet very well defined. None of these methodologies explicitly specifies the modes of processing and the types of interface they allow. Thus, the TECHNOLOGY dimension in rather weak in all these methodologies.

4 General discussion

The GAIA, MaSE, HLIM, MMTS and AOMEN methodologies use objects as the common ground of their techniques. This brings along a major benefit to the development team: the expertise already accumulated through experience with object-oriented techniques, tools and methods can greatly facilitate integration of agent technology. CoMoMAS and MAS-CommonKADS methodologies constitute an extension of the knowledge-based method CommonKADS. The methodologies that constitute an extension of knowledge-based methods and the MMTS methodology provide models that better take into account the agents' internal states. However, these models, as well as MASB, are relatively complex. The software support for representation used in HLIM (UCMs) is very interesting for comprehension, interpretation and communication during MAS development. The formalism used by the MaSE methodology makes its components generic and facilitate verification. The dimensions of COOPERATION and ORGANIZATION still require a lot of additional work in all nine methodologies. Finally, the TECHNOLOGY dimension is not seriously taken into account by any of these methodologies and thus deserves further consideration.

5 Unification of methodologies

We here propose to look at our comparative analysis of the above nine MAS development methodologies as a first step towards their unification by combining their strong points. Such an endeavour is similar in spirit to the one that gave birth to UML. However, the challenge seems a lot more difficult as it involves far greater complexity. We make some preliminary suggestions for the design of a relatively complete MAS methodology. The unification of MAS development methodologies:

a) would lead to a better methodology that would take into account all MUCCMAS dimensions. The latter are fundamental to the development of MAS and are not, as of today, covered by any single methodology.
b) could be based, for the METHODOLOGY dimension, on the MAS-CommomKADS methodology since it appears to be the strongest along this dimension. The spiral model used by MAS-CommonKADS could be replaced by the model nabla [Kolski, 1997] since it seems to better integrate the user within the overall software development process.
c) could be based, for the REPRESENTATION dimension, on UCMs (from the HLIM methodology) since they are simple and clear for representation purposes. UCMs could even help decrease the complexity of MAS-CommomKADS models. To facilitate verification in MAS, we recommend investigating the integration of the concepts from the AgML and AgDL languages used in the MaSE methodology. Indeed, one can use the diagrams of communication to describe the conversations between agents. These diagrams use finite state machines which capture dynamics on the level of the exchange of messages between the agents. These machines lead to an algebraic specification of the conversations, which makes it possible to build formal proofs to validate the interactions between the agents and thus the MAS. Conversations between agents constitute the physical support of the co-operation and coordination in MAS.
d) could be based, for the AGENT dimension, on the MAS-CommomKADS methodology since its expertise and agent models better take into account the agents' intrinsic characteristics. Also of interest are the representation principles of agents' features in the MASB methodology which allow Dennett's three stances to be covered.
e) could be based, for the ORGANIZATION dimension, on the models of [Zambonelli et al, 2000] which, via the contribution of Jennings and Wooldridge, are linked to the GAIA methodology.

f) could be based, for the COOPERATION dimension, on the models of Cassiopeia which are very interesting for the dynamics of the interaction between agents and can help solve non cooperative states. In order to support the design of open MAS, we suggest to use the dynamic interaction model (MID) of [Ribeiro, 2000] .

It seems to us that a relatively complete methodology for MAS design and development should cover in sufficient depth all the criteria within the six dimensions of the MUCCMAS framework we proposed above. This framework also acts as the foundations of the unification scheme proposed here. What we propose is not a methodology but rather an approach to the unification of the methodologies here studied.

6 Conclusion

In this paper, we made a comparative analysis of nine well-known MAS development methodologies. Following this analysis, we proposed an initial unification scheme for these various methodologies, similarly to what has happened for UML which resulted from the unification of the OMT, Booch and OOSE object-oriented methods. Our goal is to contribute to the convergence of MAS concepts and methodologies, and to the standardization of MAS development platforms. As far as future work is concerned, we would like to look at the possibility of using our MUCCMAS grids as a tool for guiding the development of new MAS methodologies and tools. Details of our work is presented in [Sabas, 2001].

Acknowledgments

We thank all researchers who kindly provided us with documentation or information related to the work presented in this paper. For their financial support, we also thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the UQTR internal funding program (FIR).

References

[Adam & Kolki, 1999] Adam, E., Kolki, C. Étude comparative de méthodes de génie logiciel en vue du développement de processus administratifs complexes, Revue de Génie Logiciel, 1999.

[AFIA/PRC, 1998] AFIA/PRC-13 Systèmes Multi-agents, Architectures de Systèmes d'Agents : http://www-poleia.lip6.fr/~guessoum/asa.htm, 1998.

[Collinot & Drogoul, 1996] Collinot, A., Drogoul, A., and Benhamou, P. Agent-oriented Design of a Soccer Robot Team, in Proceedings of the Second International Conference on Multi-Agent systems (ICMAS-96), 41-47, Kyoto, Japan, December 1996.

[Deloach, 1999] Deloach, S.A. Multiagent Systems Engineering : A Methodology and Language for Designing Agent Systems, Proceedings of AOIS-1999, Heidelberg June 1999.

[Dennett, 1987] Dennett, D.C. The Intentional Stance. MIT Press, Cambridge, Mass., 1987.

[Ealmmari & Lalonde, 1999] Ealmmari, M., and Lalonde, W. An Agent-Oriented Methodology : High-Level and Intermediate Models. Proceedings of AOIS-1999, Heidelberg June 1999.

[Flores-Mendez, 1999] Flores-Mendez, R.A. Towards the Standardization of Multi-Agent Systems Achitectures (An Overview), ACM Crossroads' Special Issue on Intelligent Agents, Summer 1999.

[Giovanni et al, 2001] www.eurescom.de/~public-webspace/P900series/P907/papersAndPresentations/AOSE2001LNCS.pdf

[Glaser, 1996] Glaser, N. Contribution to Knowledge Modelling in a Multi-Agent Framework (the CoMoMAS Approach). PhD thesis, Université Henri Poincaré, Nancy I, France, November,1996.

[Iglesias et al,1997] Iglesias, C.A., Garjo, M., Gonzàlez, J.C. and Velasco, J.R. Analysis and Design of Multiagent Systems using MAS-CommonKADS, in AAA'97 Workshop on Agent Theories, Architectures and Languages, Providence, RI, July 1997, ATAL. An extended version of this paper has been published in Intelligent Agents IV : Agent Theories, Architectures, and Languages, Springer Verlag,1998.

[Kendall et al, 1996] Kendall, E.A., Malkoun, M.T., and Jiang, C. A Methodology for Developing Agent-based Systems for Enterprise Integration, in D. Luckose and C. Zhang, editors, Proceedings of the First Australian Workshop on DAI, Lecture Notes on Artificial Intelligence, Springer-Verlag :Heidelberg, Germany, 1996.

[Kinny et al, 1996] Kinny, D., Georgeff, M. and Rao, A. A Methodology and Modeling Technique for Systems of BDI Agents, in W. van der Velde and J. Perram, editors, Agents Breaking Away : Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World MAAMAW'96, LNAI Volume 1038, Springer-Verlag : Heidelberg, Germany,1996.

[Kolski, 1997] Kolski, C. Interfaces Homme-Machine (application aux systèmes industriels complexes), Editions HERMES, Paris 1997.

[Moulin et al, 1996] Moulin, B., and Brassard, M. A Scenario-based Design Method and an Environment for the Development of Multiagent Systems. In D. Lukose and C. Zhang, editors, First Australian Workshop on Distributed Artificial Intelligence, LNAI volume 1087, 216-231, Springer-Verlag :Heidelberg, Germany,1996.

[Pascot & Bernadas, 1993] Pascot D., and Bernadas, C. L'essence des méthodes : Étude comparative de six méthodes de conception de systèmes d'information informatisés. INFORSID'93, "Systèmes d'information, systèmes à base de connaissances", Lille, 11-14 Mai 1993.

[Ribeiro, 2000] Ribeiro, A. Un modèle d'interaction dynamique pour les systèmes multi-agents. Thèse de doctorat, Université Joseph Fourier, 2000.

[Sabas, 2001] Sabas, A. Systèmes Multi-Agents : une analyse comparative des méthodologies de développement ; Vers la convergence des méthodologies de développement et la standardisation des plateformes SMA. Mémoire de maîtrise, Département de mathématiques et d'informatique, Université du Québec à Trois-Rivières (Canada), décembre 2001.

[Wooldridge et al, 2000] Wooldridge, M., Jennings, N.R. and Kinny, D. The GAIA Methodology for Agent-Oriented Analysis and Design, Journal of Autonomous Agents and Multi-Agent Systems, 3(3) 285-312, 2000.

[Zambonelli et al, 2000] Zambonelli, F., Jennings, N.R. and Wooldridge, M. Organisational Abstrations for the Analysis and Design of Multi-Agent Systems, Proceedings 1st Int. Workshop of Agent-Oriented Software Engineering, Limerick, Ireland, 235-251, 2000.

		METHODOLOGIES								
Criteria	**Values of criteria**	**GAIA**	**MaSE**	**MMTS**	**HLIM**	**CoMoMAS**	**MASB**	**MAS-CommonKADS**	**AOMEM**	**Cassiopeia**
Nature of the agents	homogeneous	N	N	N	N	N	N	N	N	N
	heterogeneous	Y	P	Y	Y	Y	Y	Y	Y	Y
Type of agents	intelligent agents	P	P	Y	Y	Y	Y	Y	Y	Y
	interface agents	P	P	Y	Y	Y	Y	Y	Y	Y
	mobile agents	P	P	Y	P	Y	P	Y		Y
	information agents	P	Y	Y	Y	P	Y	Y	Y	P
	autonomous agents	P	P	Y	P	Y	Y	Y	Y	Y
Agents attributes	adaptability	P	P	P	P	Y	P	P	P	P
	autonomy	P	P	Y	Y	Y	Y	Y	Y	Y
	cooperative behaviour	P	P	Y	Y	Y	Y	Y	Y	Y
	inferential capability	P	P	Y	Y	Y	Y	Y	Y	Y
	communication ability	P	P	Y	Y	Y	Y	Y	Y	P
	mobility									
	personality	P	P	Y	Y	Y	Y	Y	Y	Y
	reactivity	P	P	Y	Y	Y	Y	Y	Y	Y
	temporal continuity	P	P	N			N			Y
	deliberative behaviour	P	P				N	Y	P	P
Agents attributions	physical stance	N	N		N	N	Y	P		P
	design stance	Y	Y	Y	Y	Y	Y	Y	Y	Y
	intentional stance	Y		Y	Y	Y	Y	Y	Y	Y

Table 1: General Characteristics of a System's Agents

(**Y (yes):** the methodology takes into account thisvalue. **N (no):** the methodology does not take into account this value. ***blank*:** based on available information, unable to conclude. **P (possible):** based on available information, we can deduce that the methodology could take into account this value.)