

# A Methodology to Specify Multiagent Systems

Aguilar Jose<sup>1</sup>, Cerrada Mariela<sup>1</sup>, and Hidrobo Francisco<sup>2</sup>

<sup>1</sup>CEMISID (Centro de Microcomputacion y Sistemas Distribuidos),  
Facultad de Ingeniería, Universidad de los Andes, Mérida 5101, Venezuela

{aguilar, cerradam}@ula.ve

<sup>2</sup>SUMA, Facultad de Ciencias, Universidad de los Andes, Mérida 5101, Venezuela  
hidrobo@ula.ve

**Abstract.** In this paper we present a methodology to specify Multiagent Systems, called MASINA. MASINA is based on MAS-CommonKADS; we use the models presented in this methodology to propose some extensions, modifications and substitutions allowing to describe the intelligent characteristics of an agent or group of agents, to use intelligent techniques for the accomplishment of tasks (e.g. artificial neural networks), and to specify emergent coordination approaches, among others.

## 1 Introduction

The investigation of methodologies for analysis and design of Multiagent Systems (MAS) is still in embryonic state. The existing methodologies for developing MAS are not exactly new; generally they are extensions from object-oriented methodologies or knowledge engineering methodologies, given their close relationship. The existing ones have important weaknesses and disadvantages which makes them unable for usage in complexly designed environments. In this article we propose a methodology for the specification of Multiagent Systems, named as MASINA (MultiAgent Systems in Automation). The term of Automation is due that this methodology was originally developed to solve automation problems using MAS [1, 2]. This methodology is based on MAS-CommonKADS [3], proposing some extensions, modifications and substitutions of the models that have already been defined in MAS-CommonKADS. MASINA depicts the fundamental elements in the MAS area, such as representing the notion of intelligence of an agent (intelligent agents modeling), in a collective level (swarm intelligence), the coordination mechanisms between agents (emerging planning, conflict resolution, etc.), the direct or indirect communication, along with other things. In addition, MASINA allows characterizing other aspects such as the use of intelligent techniques for carrying out tasks, the use of reference models for specifying agents and the definition of conversations between agents as well as the speech acts (interactions) within these conversations.

## 2 Multiagent Systems Specification Methodologies

There are three main groups of propositions of designing methodologies based on agents, the first one is based on object-oriented methodologies, the second one is based on

methodologies coming from the knowledge engineering and the third one is based on methodologies coming from the agent paradigm itself. There are several proposals on methodologies for object-oriented agents specification, the most important ones is presented in Wood [4]. This methodology begins with an initial specification of the system and shows a set of documents with formal design based on graphics. About the methodologies coming from knowledge engineering, there is one relevant methodology which is based on CommonKADS methodology [3]. MAS-CommonKADS is the methodology that best covers the specification elements of a MAS, nevertheless, this is not implying that it does not have weaknesses: (a) the difficulty to specify an agent as a MAS, (b) the lack of dynamic schemes management for communication and (c) the lack of learning models. Finally, the methodologies coming out of the current agents' theory are founded in a social structure, where there are individuals (agents), groups and organizations. Among these methodologies we find Gaia [5], Prometheus [6], and Tropos [7]. The common problem of these methodologies is that the analysis and/or specification phases are based on the agents' paradigm, and it is not taken into account that a general analysis model can propose a multiagent scheme that would not be the most appropriate. The main design element is the social role of the agent and not its components. Some others methodologies are proposed in [8, 9].

Particularly, the MAS-CommonKADS methodology is an extension from CommonKADS model for knowledge engineering [10], adding aspects from the object-oriented methodologies like OMT, Object Oriented Software Engineering (OOSE) and Responsibility Driving Design (RDD). It has six phases, from which seven models derive. The phases are: i) **Conceptualization**: This is the extraction and purchasing of the knowledge for obtaining a first description of the problem, ii) **Analysis**: Also called Requirements Analysis, its objective or product is the requirements list. The first six models are developed in this phase, iii) **Design**: The final model of the organization is obtained under the MAS paradigm, iv) **Coding and testing**: In this phase each agent is coded and tested, v) **Integration**: Integration is accomplished into the real platform where the system will work. Then, the whole system is proved, vi) **Operation and maintenance**: Is the working phase of the MAS, in which updating tasks are performed (maintenance) on the system's components in order to adapt it to the environmental evolution or new requirements.

This methodology uses the next seven models [3]:

- **Agent Model**: Describes each one of the agents of the MAS, its objectives, characteristics, abilities, capacities, restrictions and the services it may offer to the agents community. This model is the heart of the methodology and allows the description of the actors in the problem to be solved; allows the description of agents based on the provided services, but it does not facilitate the description of an agent as a MAS.
- **Tasks Model**: In this model the tasks that should be carried out organizational wise are described. The agent's actions are specified; this way, each agent has a bank of realizable tasks, which have very well defined properties.
- **Organization Model**: It is the way for describing the organization in which the MAS will be integrated. Particularly, it describes the organization to be modeled by the MAS, before and after implementing the MAS. This model

allows considering whether the solution to the problem is the incorporation of a MAS, so it facilitates analyzing the problem and finding a solution.

- **Communication Model:** This model describes the interactions between an human agent and a software agent and it is centered in considering human factors for such interactions. It is a model based in messages, so it only specifies pre-established direct communication; it does not allow the specification of a spontaneous communication between them. This model is not detailed in MAS-Common KADS and it is believed to be derived from a coordination model.
- **Experience Model:** This is the knowledge model. The experience model makes the difference between the knowledge of the application and the knowledge for the problem solution.
- **Coordination Model:** It describes the interactions between the software agents, the used protocols and the necessary capabilities for carrying out these interactions; the coordination model defines the technical bases for developing the communication model.
- **Design Model:** It is built from the previous models in the analysis phase. In this phase the MAS design model is created.

### 3 MASINA: MultiAgent Systems in Automation

The MASINA methodology is an extension from the object-oriented MAS-CommonKADS methodology. MASINA is based on the same development cycle of MAS-CommonKADS, but with the following main modifications and contributions:

- In the Conceptualization phase, MASINA uses UML 2.0 activities diagrams [11]. The activities diagram allows the description of the offered services by the agent, dissociating it into the needed activities for carrying it out.
- In the analysis phase, MASINA introduces four of the seven proposed models in MAS-CommonKADS, enough for describing the basic characteristics of the MAS (see figure 1): Agents Model, Tasks Model, Communication Model, Coordination Model. Furthermore, MASINA replaces the Experience Model by the Intelligence Model. Intelligent Model describes all the aspects to generate an intelligent behaviour: learning, reasoning, knowledge representation, and experience accumulation.
- The Coordination and Communication models have different meaning in MASINA. The Coordination Model describes the conversations and the Communication Model describes the speech acts. For better understanding the Coordination Model, MASINA uses the UML interaction diagrams to describe the conversations.
- Agent Model has new attributes to specify different abstraction levels in a community of agents or if the agent is designed using a development framework.
- Task Model allows specifying the tasks based on intelligent techniques (genetic algorithms, artificial neural network, etc.), and the task decomposition.

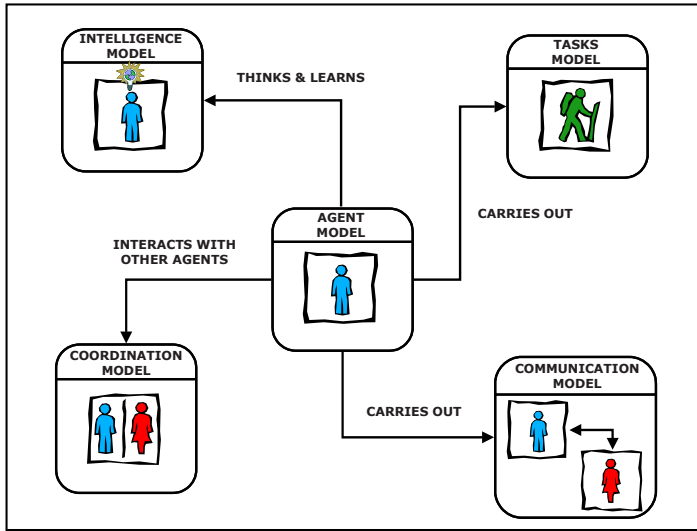


Fig. 1. MASINA Models

### 3.1 Agents Model

This model specifies, just like in the MAS-CommonKADS, the characteristics of an agent such as: abilities, services, etc. In MASINA, this model has two new attributes: (a) “agent components”, which allows to indicate whether the agent is a MAS (allows to define abstraction levels or hierarchies in the design of the MAS), and (b) “reference frame”, which indicates whether the specification of such an agent is based on an MAS reference model already existing, like the SCDA [2, 12, 13]. For defining the agent model, we use the pattern shown in table 1.

### 3.2 Tasks Model

The agents have been provided with new attributes: (a) one for specifying the properties of the tasks which require the use of intelligent techniques, and (b) another for widely describing the whole procedure to be followed for the execution of such a tasks (sub-tasks). This way, the agents activate their intelligences according to the tasks type they are carrying out (whether they are intelligent or not). For describing the tasks model, MASINA makes use of two models: one of them indicates the service-task relationship (each service has a set of tasks to be carried out and a single task could be associated to different services), the other model specifies the different tasks (see table 2).

### 3.3 Intelligence Model

The model proposed by MAS-CommonKADS only characterizes everything related to experience accumulation in an agent, but it does not take into account other elements that allow the agent to have an intelligent behavior. MASINA describes all

**Table 1.** Information for the Agent Model

AGENT	
Name	Name of the agent
Position	Location of the agent into the MAS
Components	It is specified whether the agent is composed by other agents
Reference Frame	It is specified whether it has been characterized by a reference frame for agents modelling
Agent description	Description of the agent's activities
AGENT'S OBJECTIVES	
Name	Objective's name
Description	Objective is described
AGENTS SERVICES	
Name	The name of the service offered by the agent is specified
Service description	The service is described
Type of service	The service can be internal, external or both (dual service) and it is specified
Input parameters	The necessary parameters for the service achievement
Exit parameters	The parameters obtained when the service is finished
Activation Condition	The activating conditions of the tasks associated to the achievement of the objective
Ending condition	The conditions that indicate the termination of tasks associated to the objectives achievement
Success condition	The conditions that indicate the accomplishment of the objective
Failure condition	The conditions which indicate the objective has not been accomplished
Ontology	The used ontology

**Table 2.** Tasks Model Information

NAME OF TASK	
Name	Name of the task
Objective	Objective of the task
Description	Defined objective is described
Associated services	Associated services to the objective
Precondition	Necessary conditions for task activation
Sub-tasks	The sub task to accomplish this task
NAME-OF-TASK INGREDIENT	
Name_Ingredient 1	The data/parameter necessary for task accomplishment is indicated
Name_Ingredient 2	...

the necessary aspects to incorporate the notion of intelligence into an agent. Basically, it is composed by a set of elements which correspond with human reasoning. It specifies the knowledge domain and the strategies and tasks the agent carries out, the experience and the knowledge associated to the problems, the reasoning methods needed for the agents to accomplish their objectives, as well as their learning mechanisms. The proposed intelligence model is presented in figure 2, with its attributes, relationships between them, and their interrelationship with the MASINA models. This model integrates concepts such as experience, knowledge representation, learning mechanism, and reasoning mechanism.

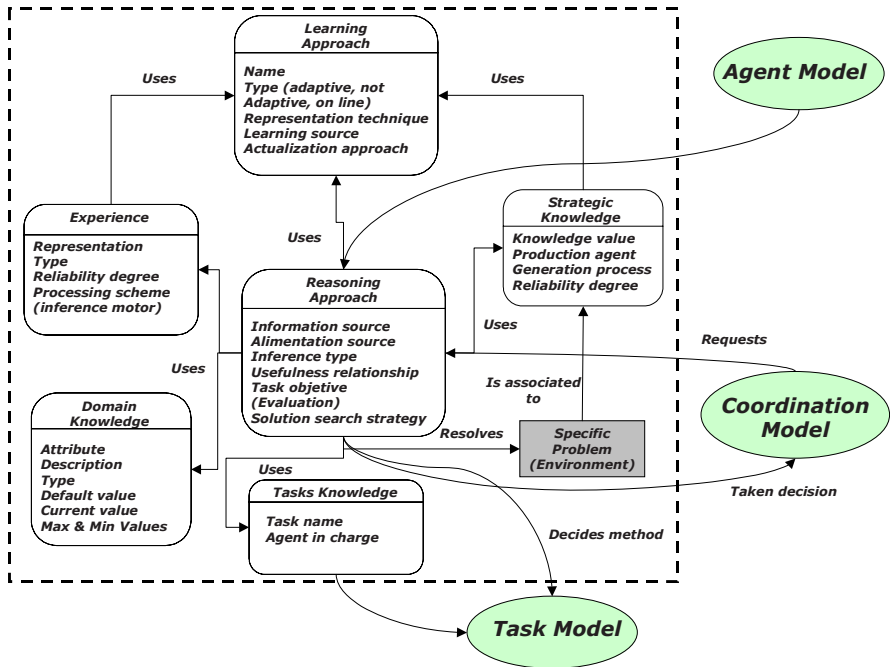


Fig. 2. Intelligence Model

### 3.4 Coordination Model

In MASINA the coordination model allows the specification of the conversations between agents. Unlike MAS-CommonKADS, the Coordination Model in MASINA is focused on the conversations allowing a coordinated communication between agents and not on the speech acts specifically involved, which are detailed in the Communication Model in MASINA. The proposed coordination model allows to the user establishing a set of strategies which would be used by the agents' communities for the objective achievement. These strategies, generally, imitate attitudes of human groups under a coordination system, such as the contracting processes and auctions, amongst others, where the interaction and information exchange generate individual actions which added up allow the reaching of the specific goal. The interactions

(speech acts) into a conversation between agents, are presented, at a graphic level, through an UML interaction diagram and such a speech acts are detailed in the Communication Model. In the Coordination Model, two types of communication between agents can be established: a direct one, based on the message exchange and the use of ontologies, and an indirect one, based on strategies of shared memory and stimulus-response methods. It is for this reason we need more detailed description of the Coordination Model in each conversation (see figure 3). In this model, the coordination scheme, the planning and communication mechanisms (direct or indirect), the meta-language and the ontology are specified.

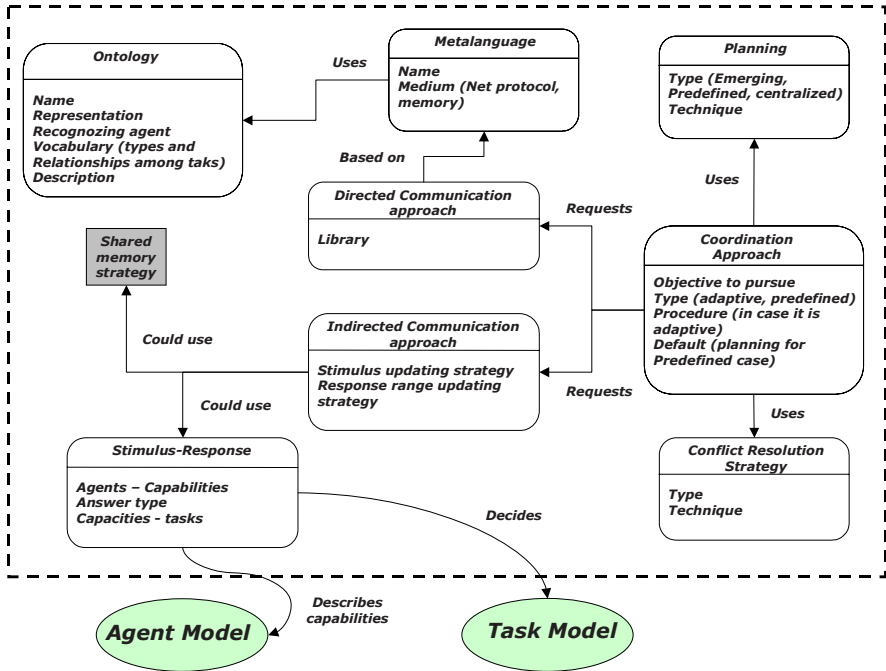


Fig. 3. Coordination Model

### 3.5 Communication Model

MAS-Common KADS does not describe this model in a detailed way, instead it is thought to be derive from the coordination model; describes the interactions between a human agent and a software agent and it is centered in considering human factors for such interaction. In our MASINA methodology, the interactions are considered in a wide way and it is proposed a Communication Model which describes the speech acts involved in the conversations between the agents. The model describes all the direct communications through the information exchange using mechanisms of the Message Exchange, and indirect communications through the information storage in objects, which resemble the interactions between agents. The attributes of the Communication Model allow the specification of a speech act in a conversation (see table 3).

**Table 3.** Communication Model Information

NAME OF THE SPEECH ACT	
Name	Specifies the name of the speech act
Type	Indicates the request characteristics (information, processing, and other requirements)
Objective	Objective of the interaction
Participating agents	The sender and receiver agents in the speech act are indicated
Initiator	The agent initiating the interaction
Exchanged Data	The data exchanged during the speech act
Precondition	The initiating conditions of the speech act
Termination Condition	The conditions determining the ending of the interaction between two agents
Conversation	Conversations where the described speech act can be found
Description	The speech act is described

## 4 Previous MASINA Applications

In [2] the proposed MASINA methodology has been used for specifying a Reference Model for Intelligent Distributed Control Systems based on agents (SCDIA). SCDIA is conformed by a community of agents and the system information and control tasks are distributed among them. This community has been called Control Agents Community: control agent, coordination agent, action agents, specialized agent and observer agent. All the SCDIA agents have been specified using MASINA.

In [1] MASINA has been used for designing an Industrial Automation Architecture based on MAS, called SADIA (Intelligent Distributed Automation System based on Agents). SADIA is an extension of the SCDIA, which includes all the automation tasks: failures management, planning and production management tasks, processes control and management of abnormal situations. As a part of the SADIA design, the SCDIA reference model, specified using MASINA, has been used for modeling the Failures and Abnormal Situations Management Agents [12, 14]. In [13] a software component allowing the implementation of a code generation system for the control agents of the SCDIA is developed. This includes the source code generation of the agent, its compilation and incorporation to the SCDIA. The system has three agents: Central agent, Code Generator agent and Behavior agent which have been specified using MASINA.

In [15] MASINA has been used for designing a Middleware based on agents. The Middleware based on agents can be used on any computational platform which stands the execution of MAS; it provides access and hardware resources management services, data and agents application. This agents-based middleware has qualities associated to the distributed systems such as interoperability, migration, security, naming, etc. In [16] MASINA has been used for modeling the Architecture of a Web Operative System (SOW), as well as for the detailed design for each one of the sub-systems composing the Operative System, which are: Communities Manager System, Replac-



ing Manager System, Web Objects Manager System and Resources Manager System. Finally, in [17] a formal method has been proposed for verifying the models that have been specified using MASINA. This method allows the validation of the proposed design for an application based on MAS.

## 5 Conclusions

The objective of this paper was the presentation of the MASINA methodology, which modifies the models that have been already proposed in MAS-CommonKADS. Particularly, MASINA modifies the meaning of the coordination and communication models and replaces the experience model by the intelligence model. This way, MASINA strengthens the communication between agents, by using indirect communication methods and asynchronic coordination schemes in order to improve the description of the conversations between agents. Furthermore, MASINA allows the use of emerging coordination schemes and the modeling of intelligent agents using the intelligence model, amongst other things. MASINA allows a better description/specification of multiagent systems. MASINA has a method that allows the verification of the proposed design for a determined application [5]. Currently, we are developing an editing tool for the MASINA models, which would permit automatically the agents code generating on JADE language.

## References

1. J. Aguilar, C. Bravo and F. Rivas. Diseño de una Arquitectura de Automatización Industrial basada en Sistemas Multiagentes. *Revista Ciencia e Ingeniería, Facultad de Ingeniería*, 25(2): 75–88, Julio 2004.
2. J. Aguilar, M. Cerrada, F. Hidrobo, G. Mousalli, and F. Rivas. “A Multiagent Model for Intelligent Distributed Control Systems”, *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Vol. 3681, pp. 191-197, 2005.
3. C.A. Iglesias, M. Garijo, J. C González, and J.R. Velazco. “Analysis and Design of Multi-Agent Systems using MAS-CommonKADS”. In M.P. Singh, A.S. Rao, and M. Wooldridge, editors, *Intelligent Agents IV. Agents Theories, Architectures and Language*, Springer, pp 1–16, 1999.
4. M. F. Wood and S. A. DeLoach. “An Overview of the Multiagent Systems Engineering Methodology”. In *First international worksho, on Agent-oriented software engineering*, Springer-Verlag, pp 207–221, 2000.
5. F. Zambonelli, N. R. Jennings, and M. Wooldridge. “Developing Multiagent Systems: The Gaia Methodology”. *ACM Trans. Softw. Eng. Methodol.*, 12(3): 317–370, 2003.
6. L. Padgham and M. Wimikoff. “Prometheus: A Methodology for Developing Intelligent Agents”. In P. Giunchiglia, J. Odell, and G. Weiss, editors, *Agent-Oriented Software Engineering III*, pp 174–185. Springer, LNCS 2585, 2003.
7. P. Bresciani, A.Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. “Tropos: An Agent-Oriented Software Development Methodology”. *Autonomous Agents and Multi-Agent Systems*, 8(3): 203–236, 2004.
8. S. Vafadar, A. Barfouroush, and M. Reza A. Shirazi. “Towards a More Expressive and Refinable Multiagent System Engineering Methodology”. In *AOIS*, pp 126–141, 2003.

9. F. Alonso, S. Frutos, L. Martinez, and C. Montes. "SONIA: A Methodology for Natural Agent Development". In *Fifth International Workshop Engineering Societies in the Agents World*, Toulouse, France, October 2004.
10. Walter van de Velde Andre Valente, Joost Breuker. The CommonKADS Library in Perspective. *International Journal of Human-Computer Studies*, 49: 391–416, 1998.
11. J. Rumbaugh, I. Jacobson, G. Booch. *The Unified Modeling Language Reference Manual*, Object Technology Series, Addison-Wesley, 1998.
12. J. Aguilar, J. Cardillo, M. Cerrada, and R. Faneite. "Agents-Based Design for Fault Management Systems in Industrial Processes", Accepted for publication, *Computer in Industry*, 2006.
13. J. Aguilar, M. Cerrada, F. Hidrobo, F. Rivas, and W. Zayas. "Development of a Code Generation System for Control Agents", *WSEAS Transactions on Computers*, Vol. 5, No. 10, pp. 2406-2411, 2006.
14. J. Aguilar, C. Bravo, E. Colina, and F. Rivas. "Sistema Multiagentes para Tratamiento de Situaciones Anormales en Procesos Industriales". In *V Congreso Nacional de la Asociación Colombiana de Automática*, pp. 24–28, Colombia, 2003.
15. J. Aguilar, V. Bravo, F. Rivas, and M. Cerrada. "Diseño de un Medio de Gestión de Servicios para Sistemas Multiagentes". In *XXX Conferencia Latinoamericana de Informática*, pp 431–439, Arequipa, Perú, Octubre 2004.
16. J. Aguilar, E. Ferrer, N. Perozo, and J. Vizcarrondo. "Architecture of a Web Operating System based on Multiagent Systems", *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Vol. 3681, pp. 700-706, 2005.
17. J. Aguilar, N. Perozo, and J. Vizcarrondo. "Definition of a Verification Method for the MASINA Methodology", *International Journal of Information Technology*, Springer-Verlag, Vol. 12, N.3, pp. 121-131, 2006.