

Simulation and Analysis of Virtual Organizations of Agents

Elena García, Virginia Gallego, Sara Rodríguez, Carolina Zato, Juan F. de Paz, and Juan Manuel Corchado

Abstract. Nowadays there is a clear trend towards using methods and tools that can help to develop multiagent systems (MAS). Thanks to the contribution from agent based computing to the field of computer simulation mediated by ABS (Agent Based Simulation) is obtained benefits like methods for evaluation and visualization of multi agent systems or for training future users of a system. This study presents a multiagent based middleware for the agents behavior simulation. The main challenge of this work is the design and development of a new infrastructure that can act as a middleware to communicate the current technology in charge of the development of the multiagent system and the technology in charge of the simulation, visualization and analysis of the behavior of the agents. The proposed middleware infrastructure makes it possible to visualize the emergent agent behaviour and the entity agent in a 3D environment. It also allows to design multi-agent systems considering organizational aspects of agent societies.

Keywords: Multiagent systems, Simulation, JADE, Repast.

1 Introduction

The contribution from agent based computing to the field of computer simulation mediated by ABS (Agent Based Simulation) is a new paradigm for the simulation of complex systems that require a high level of interaction between the entities of the system. Possible benefits of agent based computing for computer simulation include methods for evaluation of multi agent systems or for training future users of a system [6]. Many new technical systems are distributed systems and involve complex interaction between humans and machines, which notably reduce their usability. The properties of ABS makes it especially suitable for simulating this

Elena García · Virginia Gallego · Sara Rodríguez · Carolina Zato · Juan F. de Paz
Juan Manuel Corchado

Computers and Automation Department, University of Salamanca, Salamanca, Spain
e-mail: {elegar, sandalia, srg, carol_zato, fcofds, corchado}@usal.es

kind of systems. The idea is to model the behaviour of the human users in terms of software agents. However, it is necessary to define new middleware solutions that allow the connection on ABS a simulation software.

This paper describes the results achieved towards a multiagent-based middleware for the agents' behavior simulation. The middleware, called MISIA (*Middleware Infrastructure to Simulate Intelligent Agents*), allows simulation, visualization and analysis of the agent' behavior. MISIA makes use of technologies for the development of multiagent systems known and widely used, and combines them so that it is possible to use their capabilities to build highly complex and dynamic systems.

The article is structured as follows: Section 2 makes a review of the most important requirements and the reasons that led to the realization of this research. Sections 3 introduces a description of the middleware specifically adapted to the simulation of virtual organizations within dynamic environments. Finally, some results conclusions are given in Sections 4 and 5.

2 Background and Requirements

Nowadays, MAS (Multi-Agent Systems) are widely used in various fields due to their inherent properties such as autonomy, local view, decentralization, coordination or cooperation among agents to solve a general problem. MAS based on organizational concepts coordinate and exchange services and information; they are capable of negotiating and coming to an agreement; and they can carry out other more complex social actions. At present, research focusing on the design of MAS from an organizational perspective seems to be gaining most ground. The emergent thought is that modeling the interactions in a MAS cannot be related exclusively to the actual agent and its communication capabilities; instead, organizational engineering is necessary as well. The concepts of rules, norms and institutions [8] and social structures [18] are rooted in the idea of needing a higher level of abstraction, independent from the agent, that explicitly defines the organizations in which the agents reside. MAS developers have focused their efforts on the organizational aspects of agent societies, using the concepts of organization, norms, roles, etc. Virtual organizations (VO) [9] are a means of understanding system models from a sociological perspective. VO have been usefully employed as a paradigm for developing agent systems [9]. One of the advantages of organizational development is that systems are modeled with a high level of abstraction, so the conceptual gap between real world and models is reduced. Also this kind of system offers facilities to implement open systems and heterogeneous member participation.

There are several different organizational approaches and platforms: JADE [17], S-Moise+ [13], RETSINA [11], Jack [12], EIDE [7], RICA-J [20], JASON [2], SIMBA [3], THOMAS [4]. However, the designers must implement all the features of simulation, if required; for example, in cases where it is necessary to take into account for both microscopic features, such as specifying the

language between agents, as macroscopic characteristics that make up the target in a simulation.

On the other hand, thanks to the contribution from agent based computing to the field of computer simulation mediated by ABS is obtained benefits like methods for evaluation and visualization of multi agent systems or for training future users of the system [4][21]. Mainly there are two ways for visualizing multiagent systems simulation: the agents interaction protocol and the agent entity. In the former, it is visualized a sequence of messages between agents and the constraints on the content of those messages. On the other hand, the latter method visualizes the entity agent and its iteration with the environment. Most software programs, such as JADE platform [1][17] and Zeus toolkit [5], provide graphical tools that allow the visualization of the messages exchanged between agents. The toolkits MASON [14], Repast [15][19] [16]and Swarm [22] provide the visualization of the entity agent and its interaction with the environment. There are other works like Vizzari et al. [23] where is developed a framework supporting the development of MAS-based simulations based on the Multilayered Multiagent Situated System model provided with a 3D visualization.

It is necessary a software that allows to join characteristics of multi-agent systems and agent-based simulations systems. It is also fundamental in the field of ABS considering the agents from the viewpoint of cooperative, where becomes more important the purpose of the society and the rules and regulations that govern and control the behavior of its entities. It is interesting to have a platform of agents that allows to design multi-agent systems with the possibility of simulation and analysis of results, and moreover provides a module to define VO. The platform, therefore, must support frames coordination between agents, in addition to being able to dynamically adapt to changes in its structure, goals or interactions.

3 MISIA Adapted to VO

The platform that meets the above requirements is called MISIA (*Middleware Infrastructure To Simulate Intelligent Agents*) [10]. It is a platform that acts as middleware for simulation and visualization of multi-agent systems. It results from the union of two existing agent platforms: JADE, a platform widely used that comply with the FIPA standards, and Repast, used in the field of Agent Based Simulation. In addition to the union of both platforms, includes an additional module that will define Virtual Organizations of agents.

The main concept introduced in this study is the notion of time in JADE, which means that the events that occurred on this platform, such as messages between agents, will be uniquely sorted thanks to the global time provided by Repast. A schematic approach of the platform shown in the figure below:

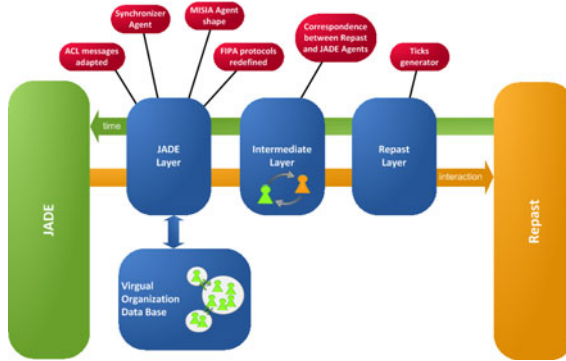


Fig. 1 MISIA Functional Approach

MISIA consists of four functional blocks:

1. *Jade Layer*. Its overall purpose is to adapt JADE to the simulation time. To this end, certain changes are required on the platform, such as: (i) designing a new template agent, which inherently manages time of the simulation; (ii) the adaptation of ACL messages containing current time information found in the simulation; (iii) a mechanism (synchronizer agent) to act as overall platform clock (in charge of the redistribution of-time events from Repast to JADE and of the notification to all JADE agents in terms of total ordering; (iv) or redefinition of the FIPA protocols provided by JADE which allow the programmer forgets tasks such as message handling and continues using these code templates in the new platform but, with the notion of time [10].

2. *Repast Layer*. It is a layer in direct contact with Repast. It is primarily responsible for the control of time of the platform: when an event goes by, it notifies the agent JADELayer Synchronizer. In this module is where all agents represented in Repast inhabit .

3. *Intermediate Layer*. It is the union of the two adjacent layers. In addition to allowing the flow of information from one layer to another, manages the correspondence between agents of the type MISIA Agent of the JADE Layer and Layer Repast representatives. While an agent's reasoning and its communication properties will reside in the mold MISIA Agent, representation in the simulation is delegated to agents living in Repast Layer, so that an agent is divided in its logic and its figure.

4. *Virtual Organization Data Base*. It is a database connected to the platform that stores all the information about a virtual organization. This includes agents, work units, services, roles and societal norms.

- 4.1. *Agents* stores all the agents that are part of the virtual organization.

- 4.2 *Work Units* are the different groups that compose the VO. The platform is capable of managing any hierarchy of units no matter how many predecessors have a particular grouping.

4.3. *Services* acts as DF (Directory Facilitator) of a multi-agent platform with FIPA specification: it records all the services offered by agents.

4.4 *Roles* defines the different roles an agent can take. There is a direct relationship between the roles and services because for that an agent can play a certain role, it is necessary to offer services that role imposes by definition.

4.5 *Societal Norms* establishes rules that govern society. There are preset rules already on the platform, they are a series of rules that must be fulfilled by any VO, as is the fact that when an agent logs into a virtual organization must be clearly identified, or that is to say , must belong to a Work Unit, and must play a certain role; or the fact that only an agent who offers all services that a role requires, can adopt that role during his lifetime. The syntax of the rules that define the end user as well as compliance with them is delegated to the case study in particular. The objective of this idea is not to restrict the type of rules can be defined with a pre-defined syntax, leaving it open to increase the range of possibilities.

This module, in addition to serving as a database, ensures that these standards inherent to the platform are met at all times.

The access to the database of the virtual organization is done through an interface that provides the platform, that is, OVMisiaAgent agent services, manager of the same. These services allow all queries and modifications that obey the established rules. Following this scheme are clear potential bottlenecks: large numbers of requests for services of the same agent. In following sections, it proposes a scheme followed in a case study, which reduces this problem, through load sharing by means of permits, and intended to serve as a general model, regardless of the purpose of case study.

4 Experimental Results

It has been developed a case study to create a multiagent system aimed at facilitating the employment of people with disabilities, so it is possible to simulate the behavior of the agents in the work environment and observe the agents actions graphically. Every job is composed of a series of tasks. Agents representing the workers have to do them, and according to their capabilities, carry out the assignment with varying degrees of success. The main objective of this application is the search for the optimal arrangement for employees to achieve greater labor productivity in a given environment. And which would also serve as a prolepsis of the reality of situations, helping to improve the employability of people with disabilities.

The application is modeled as a multi-agent system where each element is represented as an agent: employees, jobs, architectural barriers, and so on. The office environment is simulated three-dimensional way with Unity 3D¹ engine, and allows to configure different architectural barriers, like a broken fire alarm, the door of an office too narrow, inclined ramps or a elevators broken.

¹ <http://unity3d.com>

The messages exchanged between different agents in the platform are translated into interactions, movement, or specific actions in the three-dimensional environment, according to the content of the message.

Therefore, the communication between the agent platform and Unity must be bidirectional: on the one hand, the creation of agents, or interactions between them must be notified from the multi-agent platform to simulation environment of Unity. On the other hand, when setting up an architectural barrier, it is necessary to notify of this change of state to the agent that represents it in MISIA. In the latter case it is necessary to send an event with source in Unity and destination in MISIA. This communication is implemented using sockets. Specifically, two TCP (Transmission Control Protocol) connections, one in each direction, following the client-server scheme, with both servers concurrently.

Because programming languages are different for each platform (MISIA uses *Java* and, in Unity, programming is done using scripts in *C#* and *Unity-JavaScript*), framework JNI (*Java Native Interface*) is used to enable this language translation. Because the volume of information exchanged between the two technologies are quite diverse (creation of agents, deletion of agents, interaction between different roles, etc.), it was necessary to design a communication protocol between MISIA and Unity that facilitates the interpretation of the information exchanged.

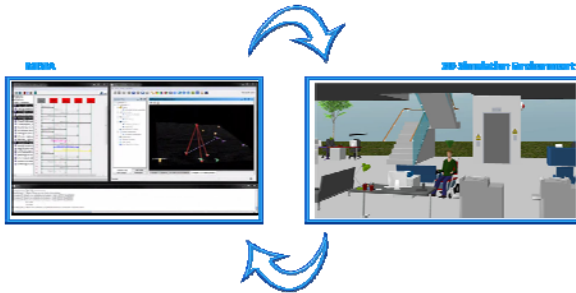


Fig. 1 Case Study: Accessibility in an Office Environment

The system is modeled as a VO that resembles reality, where the work units will be the departments that make up the office. It has three main Work Unit: the Quality Department, Human Resources Department and Production Department. Production department is subdivided according to the different occupations available in the virtual office, and the Quality Department has a sub-department where they are located all barriers. As shown in Figure 3, the topology followed for the design of the VO is a federation, as it is the best structure for the case study (a virtual office): each unit has an agent that represent it, which is the contact with other organizational units. This maintains a hierarchical structure by role of each employee.

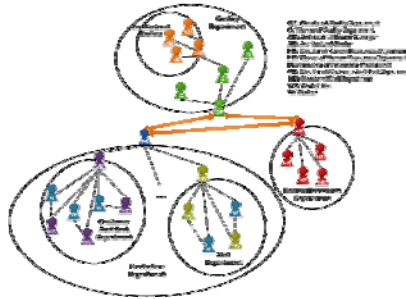


Fig. 2 Virtual Organization Structure

In the Production Department will be placing all worker agents and the agents representing jobs. To maintain a complete record of all activity performed by a worker, each time you perform a specific task, communication is maintained with their respective job (exchange of ACL messages according to the protocol FIPA more suitable in every situation). So so you can get information from the time the employee took to perform a given task and the start and end of the same.

4.1 Control Access to the VO

One of the most common problems that a dynamic VO always has to face is the behavior that it must adopt when an external agent appears. The agent's benevolence can be supposed but really, the cooperation is not guaranteed. It is needed to maintain a strict control of the agents that belong to the VO. The Human Resources Department of the organization's structure is in charge of this management that covers the entrance and exit of agents and the fulfillment of the norms defined by the end user.

In previous sections, the existence of the OVMisiaAgent has been mentioned. This agent is in charge of offering the services to interact with the database, which store the information related to the VO. Another one of the inherent norms of a VO included in MISIA is that an agent that does not own an specific authorization cannot access directly to the database. OVMisiaAgent owns this authorization from the beginning, reason why it can accede to the database and is enabled to offer all the management services related to database. This agent can give authorizations to other agents, so that they also can access to the database, without having to consume the services of OVMisiaAgent. This means that the charge can be distributed in a personalized way, which is the function carried out by the Human Resource Agents. These are agents playing a role that can own authorizations from OVMisiaAgent to access directly to the database. At any moment, indeed with simultaneous access, the consistency of the database is guaranteed. Moreover, this agents can define their own service interface allowing the fulfillment of the norms that have been previously defined in the case study.

In this particular case study, there are two different roles at the Department of Human Resources: Human Resources Worker (HR) and Human Resources Department Director (HB). HB keeps track of the amount of free jobs positions.

When an unemployed agent applies for a job, it communicates with a HR, and the HR first checks if he has the necessary skills (implements every service needed to adopt the role that it wants). Then, it asks the HB for any vacancy in the job that the unemployed is applying for. If everything is correct and the job can be assigned to him, HR will report to the new employee, giving him the information about his new job position and adding him into the VO. The agents that continuously are interacting with the database are provided with the specific authorization: insertion of new agents (since in this case, the agents with the Employer role are not allowed to make modifications in the Agents table of the database, which is a norm defined previously in the case study and the RH agents must guard by its fulfillment all time), verification of its services, if those services already exist in the database or if all the services required by the role that wish to adopt are implemented. This way, the required cost to make requests to OVMisiaAgent is reduced to half regarding to the number of interchanged messages.

4.2 Learning in an VO

The main goal of Quality Department in this study case is to observe worker agents checking the time they spend to perform their tasks according to the architectural barriers that have affected them during the execution of that job. They are agents whose main goal is to learn depending on what they see. In this situation, in Quality Department, there are two main roles: Director of Quality Department (QB) and Worker of Quality Department(Q). The performance of this team is as follows: when a new employee joins the Virtual Organization, HR agent who gave him the job notifies the QB agent that a new employee exists and Quality Department must observe him. QB delegates this task to one of their employees, Q agents. At the beginning, there isn't any Q agent, but QB creates them depending on the number of employee agents in the Virtual Organization they have to study. Thus, this department adapts to the size of the Virtual Organization. Q agents tasks are to poll time to time to the employees who have been assigned to them, to know what tasks they are doing and what troubles they have when they execute their job, collecting details of every activity, and thus, future conclusions can be drawn by Quality Department, so they can readapt employees to different workstations if work efficiency can be improved with this decision.

This same scheme can be used in other kind of Virtual Organizations, when the goal of the study case is not as important as the capacity of this model to collect information in a decentralized way.

5 Conclusions and Future Works

The simulation of multiagent systems in several levels of details and the emergent behavior is fundamental for analyzing the systems processes. This study showed in detail the visualization and simulation infrastructure for developing the MAS behavior simulators. MISIA allows simulation, visualization and analysis of the behavior of agents. With the MAS behavior simulator it is possible to visualize the emergent phenomenon that arises from the agents' interactions.

Certain systems such as the case study presented in this article require well-defined topology. The aim of the simulations is to get more realistic, and this idea is linked to a great number of factors to consider. With the possibility of design of virtual organizations, we pretend reinforce the platform with this extra factors, approaching terms of social simulation. The agent representation is doing more photo-realistic: there is a 3D agent visualization in more levels of details showing the interaction them would make the system complete and realistic. Moreover, the interactivity with the user is allowed: the user can visualize the agent state and its simulation individually. The platform improve the interactivity by means of allowing the interaction of the specialists with the live execution besides the basic functionalities such as play, pause, stop and increase/decrease the speed, by means of putting some substances in the position and observing the emergent behavior. As a future line of work, we are considering the generation of reports about the information visualized during the simulation process in several levels of detail, which could increase the comprehension about the process. MISIA is the ideal framework for this purpose.

Acknowledgments. This work has been partially supported by the MICINN project TIN 2009-13839-C03-03.

References

- [1] Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: Jade a white paper. *EXP in Search of Innovation* 3(3), 6–19 (2003)
- [2] Bordini, R.H., Hübner, J.F., Wooldridge, M.: *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley & Sons, Ltd. (2007)
- [3] Carrascosa, C., Rebollo, M., Soler, J., Julian, V., Botti, V.: SIMBA Architecture for Social Real-Time Domains EUMAS 2003: 1st E. In: *Workshop Multi-Agent Systems* (2003)
- [4] Carrascosa, C., Giret, A., Julian, V., Rebollo, M., Argente, E., Botti, V.: Service Oriented MAS: An open architecture. In: Decker, Sichman, Sierra, Castelfranchi (eds.) *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Budapest, Hungary, May 10-15, pp. 1291–1292 (2009)
- [5] Collis, J.C., Ndumu, D.T., Nwana, H.S., Lee, L.C.: The zeus agent building tool-kit. *BT Technol. Journal* 16(3) (1998)
- [6] Davidsson, P.: Multi Agent Based Simulation: Beyond Social Simulation. In: Moss, S., Davidsson, P. (eds.) *MABS 2000. LNCS (LNAI)*, vol. 1979, pp. 97–107. Springer, Heidelberg (2001)
- [7] Esteva, M., Rodríguez-Aguilar, J.-A., Sierra, C., Garcia, P., Arcos, J.-L.: On the Formal Specification of Electronic Institutions. In: Sierra, C., Dignum, F. (eds.) *AgentLink 2000. LNCS (LNAI)*, vol. 1991, pp. 126–147. Springer, Heidelberg (2001)
- [8] Esteva, M.: *Electronic Institutions: from specification to development*. Ph.D. Thesis, Technical University of Catalonia (2003)
- [9] Ferber, J., Gutknecht, O., Michel, F.: From Agents to Organizations: An Organizational View of Multi-agent Systems. In: Giorgini, P., Müller, J.P., Odell, J. (eds.) *AOSE 2003. LNCS*, vol. 2935, pp. 214–230. Springer, Heidelberg (2004)

- [10] García, E., Rodríguez, S., Martín, B., Zato, C., Pérez, B.: MISIA: Middleware Infrastructure to Simulate Intelligent Agents. In: Abraham, A., Corchado, J.M., González, S.R., De Paz Santana, J.F. (eds.) DCAI 2011. AISC, vol. 91, pp. 107–116. Springer, Heidelberg (2011) ISBN: 978-3-642-19933-2
- [11] Giampapa, J.A., Sycara, K.: Team-Oriented Agent Coordination in the RETSINA Multi-Agent System. Tech. Report CMU-RI-TR-02-34, Robotics Institute, Carnegie Mellon University, Presented at AAMAS 2002 (December 2002)
- [12] Howden, N., et al.: JACK intelligent agents-summary of an agent infrastructure. In: Proceedings of IEEE International Conference on Autonomous Agents, Montreal (2001)
- [13] Hübner, J.F., Sichman, J.S., Boissier, O.: S-Moise+: A Middleware for Developing Organized Multi-Agent Systems. In: Boissier, O., Padget, J., Dignum, V., Lindemann, G., Matson, E., Ossowski, S., Sichman, J.S., Vázquez-Salceda, J. (eds.) ANIREM 2005 and OOP 2005. LNCS (LNAI), vol. 3913, pp. 64–78. Springer, Heidelberg (2006)
- [14] Luke, S., Cioffi-Revilla, C., Panait, L., Mason, S.K.: A new multiagent simulation toolkit. In: Proceedings of the 2004 SwarmFest Workshop (2004)
- [15] North, M.J., Howe, T.R., Collier, N.T., Vos, J.R.: The repast symphony runtime system. In: Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms (2005)
- [16] North, M.J., Collier Nicholson, T., Vos Jerry, R.: Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit. *ACM Transactions on Modeling and Computer Simulation* 16(1), 1–25 (2006)
- [17] JADE, Java Agent Development Platform, <http://JADE.tilab.com>
- [18] Van Dyke Parunak, H., Odell, J.J.: Representing Social Structures in UML. In: Wooldridge, M.J., Weiß, G., Ciancarini, P. (eds.) AOSE 2001. LNCS, vol. 2222, pp. 1–16. Springer, Heidelberg (2002)
- [19] Repast, http://repast.sourceforge.net/repast_3/index.html
- [20] Serrano, J.M., Ossowski, S.: RICA-J -A Dialogue-Driven Software Framework for the Implementation of Multiagent Systems. In: JISBD Taller en Desarrollo de Sistemas Multiagente (DESMA-2004), Málaga, pp. 48–61 (2004)
- [21] Shendarkar, A., Vasudevan, K., Lee, S., Son, Y.-J.: Crowd Simulation for Emergency Response using BDI Agent based on Virtual Reality. In: Proceedings of the 2006 Winter Simulation Conference, pp. 545–553 (2006)
- [22] Swarm, <http://www.swarm.org>
- [23] Vizzari, G., Pizzi, G., da Silva, F.S.C.: A framework for execution and visualization of situated agents based virtual environments. In: Workshop dagli Oggetti agli Agenti, pp. 22–25 (2007)
- [24] Wooldridge, M., Jennings, N.R.: Agent Theories, Architectures, and Languages: a Survey. In: Wooldridge, M., Jennings, N.R. (eds.) *Intelligent Agents*, pp. 1–22. Springer, Heidelberg (1995)
- [25] Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing Multiagent Systems: The Gaia Methodology. *ACM Transactions on Software Engineering and Methodology* 12, 317–370 (2003)