

Visualization of Agents and Their Interaction within Dynamic Environments

Elena García¹, Virginia Gallego¹, Sara Rodríguez¹, Carolina Zato¹,
and Javier Bajo²

¹ Computer and Automation Department, University of Salamanca, Salamanca, Spain
{elegar, sandalia, srg, carol_zato}@usal.es

² Computer Department, Pontifical University of Salamanca, Salamanca, Spain
jbajope@upsa.es

Abstract. Many new technical systems are distributed systems that involve complex interaction between humans and machines, which notably reduces their usability. The properties of Agent Based Simulation make it especially suitable for simulating this kind of system. However, it is necessary to define new middleware solutions that allow the connection of simulation and visualization software. This paper describes the results achieved from a multiagent-based middleware for the behavior simulation and visualization of agents. The middleware modules presented in this study allow a complete integration of technologies for the development of Multiagent Systems and Agent Based Simulation, the construction of virtual organizations of agents, and the connection to external modules that represent the entities of the agents.

Keywords: Multi-agent systems, Simulation, Visualization, Virtual Organizations.

1 Introduction

Agents and Multiagent Systems (MAS) are adequate for developing applications in dynamic, flexible environments. Autonomy, learning and reasoning are especially important aspects for an agent. The development of open MAS, and of Virtual Organizations (VO) [17][18][23][6] in particular, is still a recent field of the multiagent system paradigm, and its development will allow the application of agent technology in new and more complex application domains. The contribution from agent based computing to the field of computer simulation mediated by ABS (Agent Based Simulation) provides benefits such as methods for evaluation and visualization of multi agent systems, or training future users of the system [7]. Many new technical systems are distributed systems that involve complex interaction between humans and machines, which notably reduces their usability. The properties of ABS make it especially suitable for simulating this kind of system. The idea is to model the behaviour of human users in terms of software agents.

However, it is necessary to define new middleware solutions that allow the connection of ABS simulation and visualization software.

There are two ways to visualize Multiagent **System** simulation: agent interaction protocol and agent entity. In the former, a sequence of messages between agents and the constraints on the content of those messages is visualized. The latter method visualizes the entity agent and its interaction with the environment. Most software programs, such as JADE platform [2][15] and Zeus toolkit [5], provide graphical tools that allow the visualization of the messages exchanged between agents.

The toolkits MASON [12], Repast [13][16] and Swarm [19] provide the visualization of the entity agent and its interaction with the environment. Swarm [19] is a library of object-oriented classes that implements the Swarm conceptual framework for agent-based models, and provides many tools for implementing, observing, and conducting experiments on ABS. MASON [12] is a multiagent simulation library core developed in Java. It provides both a model library and an optional suite of visualization tools in 2D and 3D. Repast [16] is a free and open-source agent-based modelling and simulation toolkit. There are other studies like Vizzari et al. [22] that have developed a framework supporting the development of MAS-based simulations based on the Multilayered Multiagent Situated System model provided with a 3D visualization.

This paper describes the results achieved from a multiagent-based middleware for the behavior simulation and visualization of agents. The middleware, called MISIA (*Middleware Infrastructure to Simulate Intelligent Agents*), allows simulation, visualization and analysis of agent behavior [8]. MISIA makes use of technologies for the development of well-known and widely used Multiagent Systems, and combines them so that it is possible to use their capabilities to build highly complex and dynamic systems. Moreover, MISIA presents a reformulation of the FIPA protocol [20] used in JADE [15], achieving several advantages such as independence between the model and visualization components, improvement on the visualization component, which makes it possible to use the concept of “time”, essential for simulation and analysis of the behavior of agents, and improvements to user capabilities, including the addition of several tools such as message visualization, 2D (and 3D agents), analysis behavioral, statistics, etc [8].

The main contribution of this paper is to present the modules that form the core of the platform and allow: the integration of both technologies, the construction of Virtual Organizations of agents, and the connection to external modules to represent the entities of the agents (in particular with a three-dimension).

The article is structured as follows: Section 2 introduces a description of the modules within the middleware specifically adapted to the simulation of MAS (MISIA). Section 3 shows some experimental results and, finally, conclusions and future works are provided in Section 4.

2 MISIA Characteristics

MISIA [8] is a framework for agent simulation, the result of the symbiosis of two well-known agent platforms [15][16]. Its purpose is to achieve a complete end tool where developers can build open and dynamic systems, analyze, simulate and

visualize the behavior of a MAS design. The platforms used are: (i) JADE [15], a free environment for the development of multi-agent systems (MAS) that simplifies the implementation of MAS through a middleware that complies with FIPA specifications and has graphical tools for debugging, and (ii) Repast Symphony [16][13][14], which is also open source and is intended to the field of Agent Based Simulation (ABS).

MISIA is primarily used for the design and implementation of models for social simulation. The reasons for carrying out a simulation of this type can be either explanatory (for processes carried out in social phenomena) or predictive (providing information on evolutionary, complex and adaptive processes of the system, and enabling the study of emergent behavior of agents in such systems).

A social system can be defined as a collection of autonomous individuals, with perception of the environment and possession of their own beliefs and goals (agents). These individuals can interact with each other and communicate, either directly or through the environment, and moreover, can evolve over time. All of these factors are offered by MISIA: interaction and communication are performed through message passing between agents, in compliance with FIPA standards, as a result of the collaboration of JADE; and time is provided by Repast Symphony, where the unit of time is the *tick*.

In short, this study provides a detailed analysis of simultaneous agent interactions without a predetermined order in any simulation of agents. The objective is to obtain predictive systems models that can produce unexpected behavior and interactions not covered.

The block diagram below shows the architecture of this platform, consisting of three main modules which will be explained in detail in later subsections.

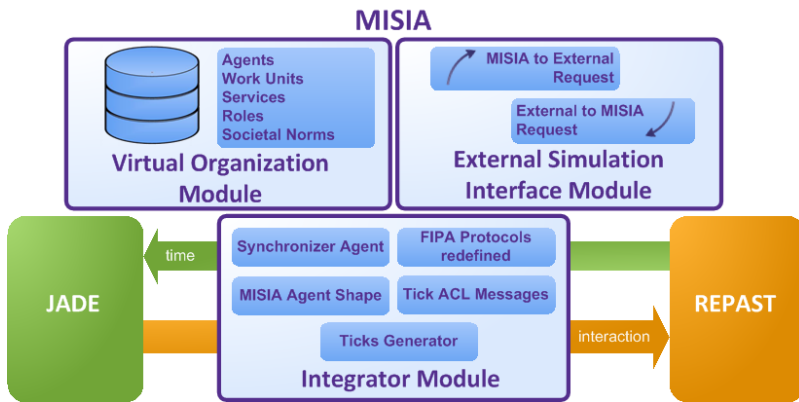


Fig. 1 Block diagram of MISIA

2.1 Integrator Module

For the combination of JADE and Repast Symphony, various components located in this module, which can be considered the core of the platform, were

implemented. The main objective of the Module Integrator is to provide developers with an API (Application Programming Interface) containing the main features of JADE and Repast adapted to the new platform:

- ACL messages adapted to the notion of time, that is, with information (on *tick-ticks*) of message delays and information at the moment that messages are sent and received;
- a "time generator" that propagates the *tickticks* generated by Repast to the rest of the platform with the aid of a *synchronizer agent*. This agent informs when a *tick* is passed to a platform agents through specific ACL messages, thus acting as a clock for all agents;
- An agent template that is derived from the template provided by JADE and includes capacities for time management. These capabilities are: an inherent behavior that can process messages received during *tick*, and process messages with time information automatically;
- and finally, a module in charge of redefining the FIPA protocols available on JADE for use with the notion of time [8].

2.2 *Virtual Organization Module*

Living together in society fosters the occurrence of unpredictable events that are not possible in isolated environments [3] [1]. Complex adaptive systems have the ability to simulate agents in their social evolution. In these simulations, the agents interact based on partial information, to achieve their individual goals, influenced by the behavior of other agents. Agents somehow compete to adapt their behavior to that of other agents, which leads to the emergence of global characteristics in the simulated model.

This module aims to give the developer the possibility of designing a society of agents, a virtual organization (VO) that allows the study of pre-specified characteristics and prediction of emergent behaviors. The main factors that characterize a virtual organization are the set of agents that are part of the organization, and the services that each of these can offer to the rest of the society, an agent communication language, a set of roles that each agent may adopt, a set of rules governing the organization, and the different subunits into which the agents are grouped as part of the organization [17]. All this information is stored in a database provided by the MISIA platform.

One of the most interesting aspects to consider of a VO is the definition of the norms [4][9] governing society. There are a series of predetermined norms in the database that are common to all VO including, for example, that every agent that exists in the organization must adopt a certain role at all times, or that an agent should always belong to a subunit of the virtual organization. However, the syntax of the norms that a developer can design for their model is limited to the particular case study. The reason for this decision is to avoid restricting the type of rules that can be defined with a predefined syntax, leaving open the possibility of increasing the range of possibilities for creating norms.

Another important aspect to consider is the regulation of access to the VO. Access to the database of the VO is governed by an interface implemented as a set of services provided by the *OVMisiaAgent*. Thus, it is possible to have access control information and the handling thereof from the VO. The main problem that a dynamic virtual organization faces is the action to be taken when an external agent appears and wants to join the organization. The benevolence of agents can be assumed, but real cooperation with other agents is not guaranteed. That is why the *OVMisiaAgent* exists, as it can enforce strict control required in virtual organization.

In order to avoid having only one agent in charge of the entire database, there is a pre-established norm in the VO: an agent with specific authorization can directly access the VO database, and is also able to generate authorizations for other agents of the VO. This is done to avoid the problems that may arise from having a single agent in charge of this functionality (such as the formation of bottlenecks due to multiple requests at successive times, or the loss of access to database if the single agent were to fall). The *OVMisiaAgent* agent is allowed to authorize other agents to distribute the charge among more entities. This simple scheme provides an optimization in the management of the OV. It is possible to create agents dynamically for this role by taking into account the number of agents that constitute the organization, and the frequency of requests to the organization's database.

2.3 External Simulation Interface Module

One of the reasons that social simulation gains importance when declaring itself a means of modeling social phenomena and a tool for social research, is that the representation of the procedures used in the simulation model are versatile. This means that the resemblance achieved in relation to the processes of the real world is quite acceptable compared to other kinds of models, such as those of a natural or even mathematical language [11]. The process of representation can become as complex as desired, making it easy to specify many details of the real process in the simulated model.

This module emphasizes this characteristic as its main goal. The ABS platforms offer many advantages and possibilities regarding the modeling and simulation of a real environment, but it sometimes needs greater versatility. Moreover, because of the actual power of graphic applications, any real situation can be modeled in as much detail as needed. As an example, the behavior of a group of predators and prey can be modeled in an ABS platform to observe the behavior of both species as a function of the number and arrangement of the animals. However, even though ABS platforms such as Repast Symphony provide 3D simulations, they are generally focused on the visualization of results and the synthesizing of huge amounts of data in 3D representations. These types of simulations are normally based on statistical factors which determine the probability of a predator getting its prey from a certain distance, for example. But the real number of variables which intervene in this decision is higher, such as the condition of the ground and its influence on the ability to reach the prey, the number and type of obstacles standing in the way, or the abilities of each species. ABS platforms are unable to handle this kind of detail, but other types of applications can model these situations in 3D whenever they arise, which means they are able to design a real life simulation.

External Simulation Interface Module is an optional interface which allows the connection between the platform agents and an external application with this type of simulation. The goal is to have agent representations in the external application, so that they can achieve clear communication with the respective platform agents.

There are four main actions which can be made concerning the agents: creation, erasure, modification and interaction. The interface offers the possibility of creating or removing an agent, both from the external application or the platform, so that results are correctly updated both in the platform or the external application. Concerning agent modification, it is possible to change a characteristic parameter of an agent, for example if the prey trips over an obstacle and its speed decreases. Finally, interaction among agents refers to defining a connection between two agents, which is normally the request of a service by one agent to another agent offering it. All these changes may have an influence on the representation of the external application, which is something to take into account. Moreover, the response of an agent to a request of its service might be influenced by the environment of the external application in which it is located at that moment.

The platform offers two kinds of interfaces for this purpose, which can be used simultaneously:

- MISIA to External Request Interface. The purpose of this interface is to ensure that the changes made in the agent platform are updated in the representation of the external application.
- External to MISIA Request Interface. Its function is to update the agent platform with all the changes made in the external application. In this case, when MISIA receives a request it is first necessary to notify the agent or agents which deal with the new information of this change in order to then update it. For this purpose, a specific agent is created to proceed with the notification of this information by means of ACL messages.

Communication between applications is carried out by means of TCP sockets: port 8898 for the first interface and port 8900 for the second interface. The basic unit of information exchanged in both directions is represented as shown in the following figure:

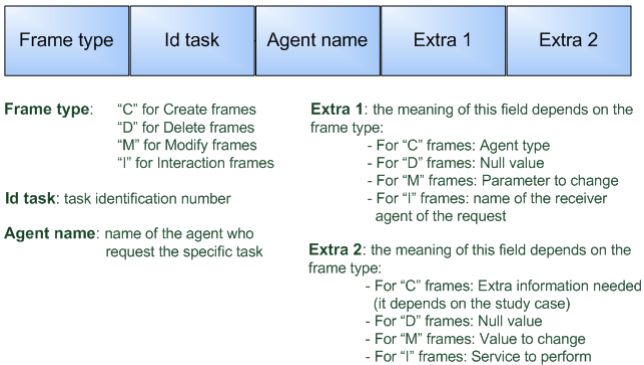


Fig. 2 Basic unit of information exchange

3 Experimental Results

In order to test the MISIA agent platform, a Multiagent System was developed to simulate an office environment and study the problems of accessibility experienced by people with disabilities in performing different jobs. The MAS is designed as a Virtual Organization similar to reality, that is, all workers, jobs and interaction elements such as architectural barriers are modeled as agents, and these are grouped into departments according to their availability and their occupation (Human Resources Department, Quality Department, Production Department with the Customer Service and Mail sub-departments). This application, which is held in MISIA, connects to another application developed with the Unity 3D engine [21] that simulates the events of the office in 3D. It also includes other features like the ability to create and delete agents, or to configure different architectural barriers from the 3D simulation application. The following figure illustrates an example of using the system and shows the interaction of agents in JADE, in REPAST and 3D application.

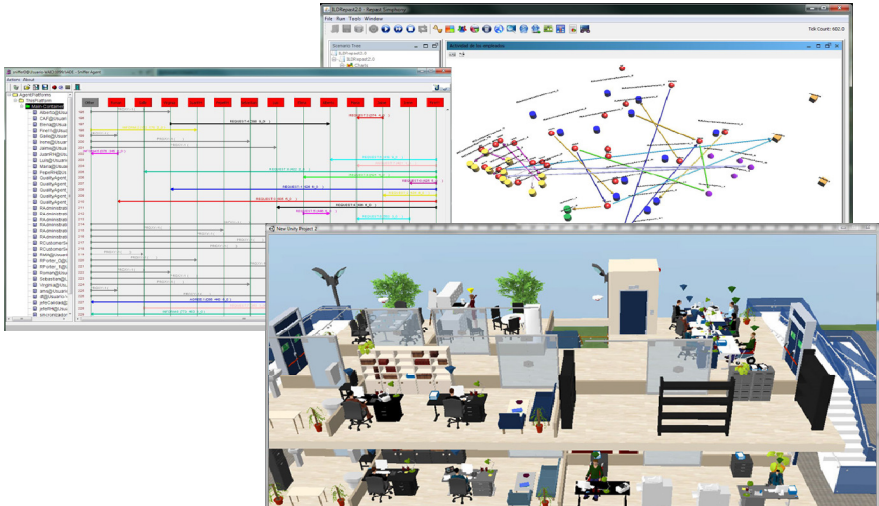


Fig. 3 Application for the study of the employment of disabled people

The main purpose of this application is to search for the optimal working conditions of the employees in the office, allowing for greater efficiency. For this purpose, several simulations of the tasks that workers have to perform, and a 3D simulation application will represent and determine the degree of success of employees in their work. There may frequently occur unusual cases in the simulations, such as a person who needs a wheelchair and cannot access the top floor of the office because the elevator is broken and no ramps are enabled; or a worker who takes a long time to perform certain tasks because the floor contains a step and accessing the destination may require a longer detour. For added versatility in

the simulations, and because the application was not dependent on the plan and the disposition of the office, the system uses algorithm A * [10] to search for the shortest path. Thus represented agents are able to find its optimal path for tasks that have to do.

The case study was modeled as a MAS, making it possible to study, at a low-level, all interactions that agents have with their environment, and to then analyze and visualize the results in Repast in order to, after many simulations, predict results. Thus, given an initial configuration for the VO agents, it is possible to predict what the optimal disposition for the work environment is. The three-dimensional simulation of the office environment here is a great incentive to make the visualization more versatile and accurate, and to provide a much more interactive interface for users of simulation application.

4 Conclusions and Future Works

One of the most important features of social simulations is that they can easily observe emergent behavior from studying models. Realistic simulations, with a significant level of detail, though complex, are best suited to represent processes that study or want to obtain an explanation of the processes, or predict outcomes. MISIA is a platform that encourages the use of complex simulations for study, and enables the analysis, simulation and visualization of both system interactions and the results obtained in a multi-agent behavior. Furthermore, the interactions between agents are well defined with the use of FIPA protocols and supported VO.

The MISIA modules presented in this study allow a complete integration of technologies for the development of MAS and ABS, the construction of virtual organizations of agents, and the connection to external modules to represent the entities of the agents. In addition, MISIA provides tools for customizing the display of the simulation, such as simulation speed, pausing, resuming, etc., which is more intuitive for an end user.

As possible future lines of work, we are considering first of all, the design of a syntax that is as generic as possible regarding the specification of the norms of a virtual organization. To achieve this goal, the syntax cannot be too restricting for the possible norms that can be defined, but must instead allow these particular norms to be specified. Moreover, reporting at different levels of detail would be an interesting aspect to consider, as it would be an incentive to understand system processes and to improve the study can be performed on the results of the simulations.

Acknowledgments. This work has been supported by the Spanish JCyL project SA225A11-2.

References

- [1] Acevedo, D.L.: Aprendizaje y comportamiento social y emergente en sociedades artificiales, Universidad Nacional de Colombia, Seminario de investigación, Maestría en Ingeniería, pp.1–9 (2007)

- [2] Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: Jade a white paper. *EXP in Search of Innovation* 3(3), 6–19 (2003)
- [3] Conte, R., Paolucci, M.: Intelligent social learning. *Artificial Society and Social Simulation* 4(1), 1–23 (2001)
- [4] Castelfranchi, C., Conte, R.: The treatment of norms in modelling rational agents: Cognitive issues. In: *Proceedings of Model Age 1995, General Meeting of Esprit Working Group*, vol. 8319, INRIA, France (1995)
- [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] Corchado, E., Pellicer, M.A., Borrajo, M.L.: A MLHL Based Method to an Agent-Based Architecture. *International Journal of Computer Mathematics* 86(10,11), 1760–1768 (2008)
- [7] 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)
- [8] 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.) *International Symposium on Distributed Computing and Artificial Intelligence. AISC*, vol. 91, pp. 107–116. Springer, Heidelberg (2011)
- [9] Hales, D.: Group reputation supports beneficent norms. *The Journal of Artificial Societies and Social Simulation (JASSS)* 5(4) (2002)
- [10] Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2), 100–107 (1968), ISSN: 0536-1567, doi:10.1109/TSSC.1968.300136
- [11] Lozares, C.: La simulación social, ¿una nueva manera de investigar en ciencia social? *Papers: revista de sociología (Ejemplar dedicado a: Visions alternatives sobre la societat i la realitat social)* 72, 165–188 (2004) ISSN 0210-2862
- [12] Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., M.: A new multiagent simulation toolkit. In: *Proceedings of the 2004 Swarm Fest Workshop* (2004)
- [13] 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* (2005)
- [14] 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)
- [15] JADE, Java Agent Development Platform, <http://JADE.tilab.com>
- [16] Repast, http://repast.sourceforge.net/repast_3/index.html
- [17] Rodríguez, S., de Paz, Y., Bajo, J., Corchado, J.M.: Social-based Planning Model for Multiagent Systems. *Expert Systems with Applications* 38(38), 13005–13023 (2011), doi:10.1016/j.eswa.2011.04.101
- [18] Rodríguez, S., Pérez-Lancho, B., Bajo, J., Zato, C., Corchado, J.M.: Self-adaptive Coordination for Organizations of Agents in Information Fusion Environments. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) *HAIS 2010. LNCS (LNAI)*, vol. 6077, pp. 444–451. Springer, Heidelberg (2010)
- [19] Swarm, <http://www.swarm.org>
- [20] Foundation for Intelligent Physical Agents. “FIPA Agent Management Specification”. Disponible en, <http://www.fipa.org/specs/fipa00001/SC00001L.html>
- [21] Unity 3D Engine, <http://unity3d.com/>

- [22] 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)
- [23] Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing Multiagent Systems: The Gaia Methodology. *ACM Transactions on Software Engineering and Methodology* 12, 317–370 (2003)