

Document downloaded from:

<http://hdl.handle.net/10251/68419>

This paper must be cited as:

Rincón Arango, JA.; Carrascosa Casamayor, C.; Garcia Marques, ME. (2014). Developing Intelligent Virtual Environments using MAM5 Meta-Model. En Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection. Springer. 379-382. doi:10.1007/978-3-319-07551-8_43.



The final publication is available at

http://link.springer.com/chapter/10.1007/978-3-319-07551-8_43

Copyright Springer

Additional Information

“The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-07551-8_43

Developing Intelligent Virtual Environments using MAM5 Meta-Model

J. A. Rincon, C. Carrascosa, and Emilia Garcia

Universitat Politècnica de València
Departamento de Sistemas Informáticos y Computación (DSIC)
Camino de Vera s/n, Valencia, Spain
{jrincon,carrasco,mgarcia}@dsic.upv.es

1 Introduction

An IVE (Intelligent Virtual Environment) is a virtual environment simulating a physical (or real) world, inhabited by autonomous intelligent entities[2]. Today, this kind of applications are between the most demanded ones, not only as being the key for multi-user games such as *World Of Warcraft*¹ (with more than 7 million of users in 2013)² but also for immersive social networks such as *Second Life*³ (with 36 million accounts created in its 10 years of history)⁴. It is in the development of these huge IVEs where the need of a quick and easy-to-use modelling toolkit arises.

Besides, Multi-Agent Systems (MAS) could be of interest to give support to these applications as a way to avoid big servers and to have some easy extensibility, scalability and fault tolerance.

This work is based on the MAM5 meta-model [1] which describes a method to design IVEs. MAM5 is based in the A & A meta-model [3] that describes environments for MAS as populated not only by agents, but also for other entities that are called *artifacts*. According to this, an IVE is composed of three important parts: artifacts, agents and physical simulation. Artifacts are the elements in which the environment is modelled. Agents are the IVE intelligent part. The physical simulation is in charge of giving the IVE the look of the real or physical world, allowing to simulate physical phenomena such as gravity or collision detection.

2 Main Purpose

In the last years, there have been different approaches for using MAS as a paradigm for modelling and engineering IVEs, but they have some open issues:

¹ <http://eu.battle.net/wow>

² <http://www.statista.com/statistics/276601/number-of-world-of-warcraft-subscribers-by-quarter/>

³ <http://www.secondlife.com>

⁴ <http://massively.joystiq.com/2013/06/20/second-life-readies-for-10th-anniversary-celebrates-a-million-a/>

low generality and then reusability; weak support for handling full open and dynamic environments where objects are dynamically created and destroyed.

As a way to tackle these open issues, and based on the MAM5 meta-model, we have developed the JACALIVE framework. It provides a method to develop this kind of applications along with a supporting platform to execute them. Figure 1 shows the steps that should be followed in order to develop an IVE according to the JACALIVE framework.

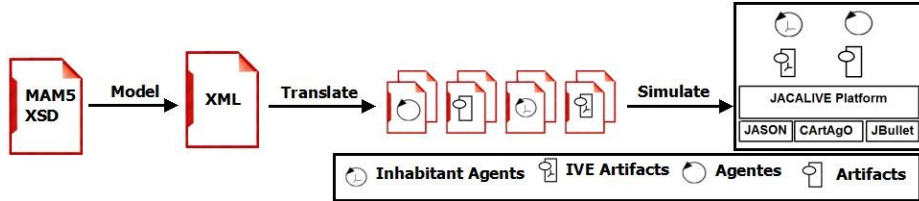


Fig. 1. General Scheme, JACALIVE

1. Model: The first step is to design the IVE. JACALIVE provides an XSD based on MAM5 meta-model. According to it, an IVE can be composed of two different types of workspaces depending on whether they specify the location of its entities (*IVE_Workspaces*) or not (*Workspaces*). It also includes the specification of agents, artifacts and the norms that regulate the physical laws of the IVE Workspaces.
2. Translate: The second step is to automatically generate code templates from design. One file template is generated for each agent and artifact. JACALIVE agents are rational agents based on JASON. The artifacts representing the virtual environment are based on CArtAgO. The developer must complete these templates and then the IVE is ready to be executed.
3. Simulate: Finally the IVE is simulated. As is shown in Figure 1, JACALIVE platform uses JASON, CArtAgO [4] and JBullet⁵. JASON offers support for BDI agents that can reason about their beliefs, desires and intentions. CArtAgO offers support for the creation and management of artifacts. JBullet offers support for physical simulation. JACALIVE platform also includes internal agents (JASON based) to manage the virtual environment.

3 Demonstration

In this section we test the versatility of the JACALIVE framework by means of the development of a case study. The selected case study is an example of modular robotics [5]. A modular robot is a self-configuring system with variable

⁵ <http://jbullet.advel.cz/>

morphology. Robots of this kind are able to adapt their shape to changes in the environment. Specifically, we want to build an IVE able to simulate virtual modular robots. These robots have the ability to change its shape depending on environment conditions. In the present example, an agent's body is initially formed by two `IVE_Artifacts`. Following the main steps of the development of this case study are summarized:

1. Model: The design of the case study is formalized using an XML based on the JACALIVE XSD. The main parts of this XML are: (i) An `IVE_Workspace` called `apodoRobot_Workspaces`. (ii) Fifteen `IVE_Artifacts`. One of the attributes of these artifacts is whether they are linkable or not, that is, if they can be joined to other artifacts or not. Three of these artifacts are linkable; two are unlinkable and the other ten form the bodies of the inhabitant agents. (iii) Five `Inhabitant_Agent`. Each one of them is associated to two of the previously defined IVE artifacts.
2. Translate: From the XML file that represents the design of the system, the JACALIVE framework automatically generates the following files: (i) Fifteen java files representing the IVE artifacts (Ten files representing the agent bodies, three files corresponding to linkable artifacts and two files correspond to unlinkable artifacts). (ii) Six JASON files that correspond to the agents. (iii) A file called `jacalive.asl`, where the developer programs the communication between agents and artifacts.
3. Simulate: Entities that have been modeled and programmed in the previous steps are simulated. Since JACALIVE physical engine handles the IVE physics simulation, any visualization engine can be used to view the simulation. In this case study the render used is implemented with OpenGL. Figure 2 shows an snapshot of the simulation.

4 Conclusions

In this paper we present a framework for the design and simulation of IVEs. This framework differs from other works in the sense that it integrates the concepts of agents, artifacts and physical simulation. Besides, IVEs developed using the JACALIVE framework can be easily modified thanks to the XML modelling and the automatic code generation.

Following the MAM5 perspective, the modules used to interact with the developed IVEs are uncoupled from the rest of the system. It allows to easily integrate different kinds of modules as needed. For example, it allows to adapt the visualization render to the requirements of the specific IVE we want to simulate.

This work is partially supported by the TIN2012-36586-C03-01, PROMETEOII/2013/019 and the FPI grant AP2013-01276 awarded to Jaime-Andres Rincon.

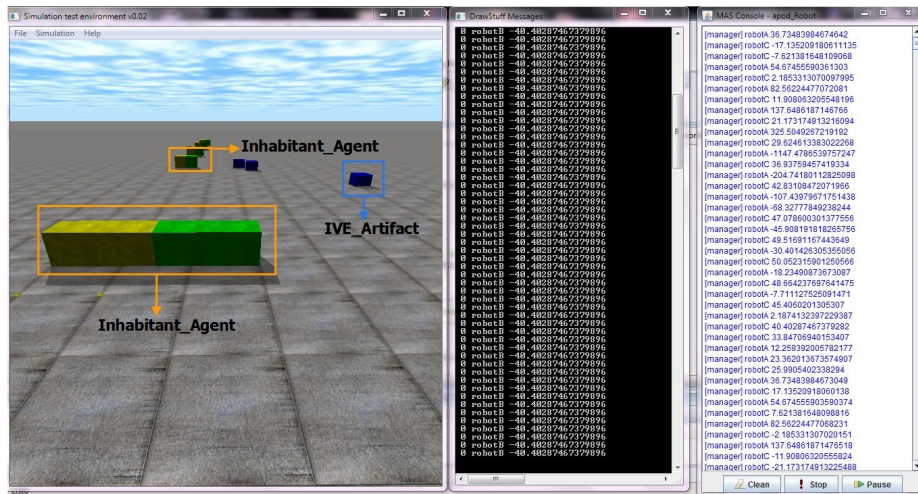


Fig. 2. Modular robotic simulation

References

1. A. Barella, A. Ricci, O. Boissier, and C. Carrascosa. MAM5: Multi-Agent Model For Intelligent Virtual Environments. In *10th European Workshop on Multi-Agent Systems (EUMAS 2012)*, pages 16–30, 2012.
2. M. Luck and R. Aylett. Applying artificial intelligence to virtual reality: Intelligent virtual environments. *Applied Artificial Intelligence*, 14(1):3–32, 2000.
3. A. Omicini, A. Ricci, and M. Viroli. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17(3):432–456, May 2008.
4. A. Ricci, M. Viroli, and A. Omicini. CArTAgO: A framework for prototyping artifact-based environments in MAS. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNAI*, pages 67–86. Springer, May 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.
5. T. Schmickl. How to engineer robotic organisms and swarms? In *Bio-Inspired Self-Organizing Robotic Systems*, page 2552. Springer, 2011.