Developing Adaptive Agents Situated in Intelligent Virtual Environments

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

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

Abstract. This paper presents a framework specially designed for the execution and adaptation of Intelligent Virtual Environments. This framework, called JACALIVE, facilitates the development of this kind of environments managing in an efficient and realistic way the evolution of parameters for the adaptation of the physical world. The framework includes a design method and a physical simulator which is in charge of giving the Intelligent Virtual Environment the look of the real or physical world, allowing to simulate physical phenomena such as gravity or collision detection. The paper also includes a case study which illustrates the use of the proposed framework.

1 Introduction

Nowadays, having software solutions at one’s disposal that enforce autonomy, robustness, flexibility and adaptability of the system to develop is completely necessary. The dynamic agents organizations that auto-adjust themselves to obtain advantages from their environment seems a more than suitable technology to cope with the development of this type of systems. These organizations could appear in emergent or dynamic agent societies, such as grid domains, peer-to-peer networks or other contexts where agents dynamically group together to offer compound services as in Intelligent Virtual Environments (IVE). An IVE is a virtual environment simulating a physical (or real) world, inhabited by autonomous intelligent entities[1].

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

\(^1\) http://eu.battle.net/wow
\(^3\) http://www.secondlife.com
\(^4\) http://massively.joystiq.com/2013/06/20/second-life-readies-for-10th-anniversary-celebrates-a-million-a/
in the development of these huge IVEs where the need of a quick and easy-to-use modelling toolkit arises.

These kinds of IVEs are addressed to a huge number of simultaneous entities, so they must be supported by highly scalable software. This software has also to be able to adapt to changes, not only of the amount of entities but also of their users needs. Technology currently used to develop this kind of products lacks of elements facilitating the adaptation and management of the system. Traditionally, this kind of applications use the client/server paradigm, but due to their features, a distributed approach such as multi-agent systems (MAS) seems to fit in the development of components that will evolve in an autonomous way and coordinated with the own environment's evolution.

This paper presents the JaCalIVE\textsuperscript{5} (Jason Cartago implemented Intelligent Virtual Environment) framework. It provides a method to develop this kind of IVEs along with a supporting platform to execute them. JaCalIVE is based on the MAM5 meta-model [2] 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.

The rest of the paper is organized as follows: Section 2 summarizes the most important related work. Section 3 describes the JaCalIVE framework. Section 4 summarizes the development process of an IVE based on modular robotics developed using JaCalIVE. Finally, Section 5 summarizes the main conclusions of this work.

2 Related work

This section summarizes the most relevant techniques and technology that the JAcalIVE framework integrates in order to design and simulate IVEs. These techniques allow JAcalIVE to develop IVEs that are realistic, complex, adaptable, and with autonomous and rational entities. First, some concepts about IVEs are presented, to continue commenting about Multi-Agent Systems concepts, as platforms and methodologies relevant to the present work. Finally, this section presents the MAM5 meta-model, as it is the starting point for the present work to model IVEs in MAS terms.

2.1 IVE

Currently, there is an increasing interest in the application of IVEs in a wide variety of domains. IVEs have been used to create advanced simulated envi-

\textsuperscript{5} http://jacalive.gti-ia.dsic.upv.es/
environments [4–6] in so different domains as education [7], entertainment [8–11], e-commerce [12], health [13, 14] and use to VR-based simulations [15].

One of the key features of any IVE is to offer a high level of user immersion. In order to achieve that, it is necessary that the IVE has the ability of simulating physical conditions of the real world such as gravity, friction and collisions. Besides, in order to increase the graphical realism, the physical simulators should include dynamic and static objects that inhabitate the IVE in a three-dimensional environment. Some of the most important developed physical simulation tools are JBullet6 and Open Dynamic Engine (ODE)7.

Another important feature of any IVE is to offer a high level of graphical realism. Currently there are in the market some well-developed graphical simulators like Unity 3D8, Unreal Engine UDK9 and Cryengine10. Although they were initially designed for videogames, they can be applied to simulate IVEs.

2.2 Multi-Agent Systems

Until now, we have highlighted the importance of giving realism to IVEs, which would enable the user to have the desired level of immersion. This realism is provided by the physical simulation and 3D visualization, but this is only one part of a virtual environment. To be an IVE, a virtual environment needs to give entities with the intelligence to enhance the user’s immersion.

MAS is one of the most employed artificial intelligence technique for modeling IVEs. This is mainly due to the characteristics that agents have, such as autonomy, proactivity, reactivity and sociability. But this does not mean that no other AI techniques can be used within MAS for IVE development. An agent can include as a decision-making mechanism other algorithms that improve the deliberative process such as reinforcement learning [16], genetic algorithms [17], markov models [18], classification[19, 20], neuronal networks [21] or use any method to hybrid artificial intelligence systems[22] etc.

However, when modeling an environment it is necessary to take into account that not all the entities are agents. The A&A meta-model [23, 24] describes a methodology for modeling environments using artifacts. Artifacts represent the first level of abstraction when modeling environments. This is mainly due to the clear differentiation of the entities which are in systems of this kind. This differentiation can determine which items are objects (Artifacts) and which are intelligent entities (Agents).

The BDI model (Belief - Desire - Intention)[25–27] is the most well-known and used agent model when designing intelligent agents. This model is based on logic and psychology, which creates symbolic representations of beliefs, desires and intentions of the agents. The beliefs are the information the agent has about

\[\text{http://jbtable.adveia/cz/}\]
\[\text{http://www.ode.org/}\]
\[\text{http://unity3d.com/unity}\]
\[\text{http://www.unrealengine.com/udk/}\]
\[\text{http://www.crytek.com/cryengine}\]
the environment. This information can be updated at each time step or not. This
obsolescence of the used information forces the agent to perform deliberative
processes. Desires are the possible actions that the agent could make. This does
not mean that every desire of an agent has to be performed. Finally, intentions
represent the actions that the agent has decided to perform. These actions may
be goals that have been delegated to the agent or may be the result of previous
deliberation processes.

Different approaches have been devised in order to develop MAS. One of the
first tools used for implementing agents is the JADE platform. JADE has been
used for the development of JGOMAS (Game Oriented Multi-Agent System
based on Jade) [28–30]. JADE does not directly provide a BDI model but there
exist an extension called JADEX allowing developers to design BDI-oriented
MAs incorporating the representation of beliefs, desires and intentions. JADEX
has been used for modeling environments like the presented in [31]. Jason is
another development tool used for MAS programming which also integrates the
BDI model.

In our proposal we employ Jason as the programming toolkit for our BDI
agents[32]. The main reason to employ JASON is its full integration with CArtAgO
(Common ARTifact infrastructure for AGents Open environments)[33]. CArtAgO
is a framework/infrastructure for modeling artifacts which can run virtual envi-
nvironments. This framework allows the implementation of open work-spaces, which
facilitates the creation of distributed environments.

2.3 MAM5

MAM5[2] is a model to design IVEs based in the A&A meta-model. It is ad-
dressed to be used by an IVE designer, that wants to design an IVE based on a
multi-agent system. As it is intended to be distributed, the human interface part
of the system is decoupled from the intelligent part, being only this last one the
part designed by means of MAM5. To have this two parts distributed facilitates
the developing, gives more flexibility to the final applications (allowing different
interfaces to be connected and at the same time) and allows to scale the final
system (thinking on massive applications with a huge number of users and/or
agents).

This model classifies the entities in the design into two different sets (as seen
in Figure 1). The first one is related to all the entities that do not have any
physical representation in the IVE (Non Virtually Physical Situated), whilst the
second one is formed by all the entities having a representation inside the IVE
(Virtually Physical Situated). Inside the former set there are Agents, Artifacts,
and Workspaces following the A&A definition. In a similar way, inside the last
set there are IVE Artifacts and Inhabitant Agents that are situated in the virtual
environment (in fact, the Inhabitant Agent will have an IVE Artifact represent-
ing its body in the IVE), and IVE Workspaces, representing the virtual place,
and the laws defining and governing such places.

MAM5 meta-model not only allows to differentiate between virtual repre-
sented entities and not virtual represented, but it also incorporates the definition
of the physical restrictions and properties in the modelling of the environment and of their inhabiting entities, respectively. That is, the designer may define the different IVE Laws governing the different IVE Workspaces in the IVE (representing the physical laws of the real world) and, he may also define the different physical properties of the entities populating such virtual environment (mass, length, ...).

3 JaCaIIVE (Jason Cartago implemented Intelligent Virtual Environment)

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: 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 JaCaIIVE framework. It provides a method to develop this kind of applications along with a supporting platform to execute them. Figure 2 shows the steps that should be followed in order to develop an IVE according to the JaCaIIVE framework.

1. Model: The first step is to design the IVE. JaCaIIVE 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 2, JaCalIVE platform uses JASON, CArtAgO and JBullet. 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.

4 Case Study

In this section a case study based on modular robots is described to show the versatility of JaCalIVE framework. Modular robots [34–36] are robots mainly characterized by their ability to reconfigure their modules and changing its shape [37, 38]. Each module of a robot is an independent entity that can be joined to other modules. This feature allows each robot to adapt its shape dynamically to changes in the environment. Currently, a wide range of domains of application are using modular robotics. For example, they are being used to search for missing persons in earthquakes [39] and to space exploration [40]. These domains need advanced virtual simulation environments like the ones MAM5 and JaCalIVE allow to test their implementations. Moreover, simulations as the one presented in this paper can be used as test beds for new adaptive algorithms, cooperative algorithms, Swarm Robotics, and so on.

Our case study implements a modular robot simulation that is composed of seven 3D modular robots models and one 3D camera model that that behave autonomously and interact with the environment. The environment is composed of a simple map without any wall. The map length and width is configured on
the modeling process, in our case is 500X500. Robots can interact among them in order to change its shape by joining other modules or environment objects. In that sense, one simple modular robot can change its shape and create a complex robot depending on the environment requirements. The case study also represent some environment entities that cannot be attached with the robots and they only represent a simple object that occupies a place in the IVE. Objects of this type are represented by fuel bladders in the 3D render.

The main steps of the development process of this case study are summarized below:

1. Model: The design of the IVE is described in terms of IVE Artifacts, Inhabitant Agents and IVE Workspaces using an XML as based on the Ja-CaIIVE XSD. The main parts of this XML are: (i) An IVE_Workspace called apodoRobot_Workspaces as show in Figure 3.

```xml
<VIRTUAL>
  <IVE_WORKSPACE NAME="apodoRobot_workspace">
  <IVE_ARTIFACTS>
    <ITEM NAME="BodyLeft"/>
    <ITEM NAME="BodyRight"/>
    <ITEM NAME="linkedArtifact"/>
    <ITEM NAME="unlinkedArtifact"/>
  </IVE_ARTIFACTS>
  <INHABITANT_AGENTS>
    <ITEM NAME="Robot"/>
  </INHABITANT_AGENTS>
  <IVE_LAWS>
    <ITEM NAME="Gravity"/>
  </IVE_LAWS>
  <IVE_WORKSPACE>
```

Fig. 3. XML that configures the IVE_Workspace

(ii) Nine 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 as show in Figure 4.

(iii) Five Inhabitant_Agent. Each one of them models one modular robot, that initially is associated to two of the previously defined IVE artifacts as show in Figure 5.

2. Translate: From the XML file that represents the design of the system, the JaCaIIVE 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 JaCallIVE 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 Unity 3D. Figure 6 shows an snapshot excerpt of the simulation.

![Image of simulation](image1)

**Fig. 6.** Excerpt of the case study simulation using Unity 3D

Figure 7 shows an example of a construction sequence to show the adaptability feature of such robots. In the sequence 1, the robot (the inhabitant agent) is displayed as a single module. This fact reduces its mobility, i.e., the robot will advance slowly. In sequence 2, an IVE Artifact has been adhered to its body changing its shape helping to improve its movement. During sequences 3, 4, 5, 6, and 7 the inhabitant agent adheres more IVE Artifacts in order to build a complex body. The final shape of the final robot allows it to improve its performance when its moving through the environment.

![Image of robotic sequence](image2)

**Fig. 7.** Modular robotic sequence
5 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 JaCaiVE 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.

To show the possibilities of such approach, a case study based on modular robotics is presented. These robots can adapt its shape to changing environment conditions. In the developed scenario, there are different modules in the environment that the agents can incorporate to their body, changing the way they move.

6 Acknowledgements

This work is partially supported by the TIN2009-13839-C03-01, TIN2011-27652-C03-01, CSD2007-00022, COST Action IC0801, FP7-294931 and the FPI grant AP2013-01276 awarded to Jaime-Andres Rincon.

References

8. Surangika Ranathunga, Stephen Cranefield, and Martin K. Purvis. Interfacing a
cognitive agent platform with a virtual world: a case study using second life. In
AAMAS, page 11811182, 2011.
9. Roberto Andreoli, Rosario De Chiara, Ugo Erra, and Vittorio Scarano. Interactive
3d environments by using videogame engines. In Information Visualisation, 2005.
computer game to be used in a real environment. In Trends in Practical Applications
11. Frank Dignum. Agents for games and simulations. Autonomous Agents and Multi-
and its application in e-commerce. In Computer Software and Applications Confer-
15. Tim Ritchings, George Anastassakis, and Themis Panayiotopoulos. Multi-agent
systems as intelligent virtual environments. In Franz Baader, Gerhard Brewka, and
Thomas Eiter, editors, KI 2001: Advances in Artificial Intelligence, number 2174
in Lecture Notes in Computer Science, pages 381–395. Springer Berlin Heidelberg,
16. Juan A. Garca-Pardo and Carlos Carrascosa. Social welfare for automatic innova-
to solve the productiondistribution planning problem for a supply chain: a genetic
18. Gra\c{c}aliz Pereira Dinuzzo, Antnio Carlos da Rocha Costa, Luciano Vargas
Gon\c{c}alves, and Alexandre Hubner. Interval-valued hidden markov models for
recognizing personality traits in social exchanges in open multiagent systems. 2008.
19. Jeff Orkin and Deb Roy. Semi-automated dialogue act classification for situated
20. Michal Wonias, Manuel Graa, and Emilio Corchado. A survey of multiple classifier
systems as hybrid systems. Information Fusion, 16:3–17, March 2014.
21. Li Jia and Miao Zhenjiang. Entertainment oriented intelligent virtual environment
with agent and neural networks. In IEEE International Workshop on Haptic, Audio
22. Emilio Corchado, Micha Wonias, Ajith Abraham, Andr C.P.L.F. de Carvalho, and
Vclav Snl. Recent trends in intelligent data analysis. Neurocomputing, 126(0):1–2,
2014. Recent trends in Intelligent Data Analysis Selected papers of the The 6th
International Conference on Hybrid Artificial Intelligence Systems (HAIS 2011)
Online Data Processing Including a selection of papers from the International
Conference on Adaptive and Intelligent Systems 2011 (ICAIS 2011).
model and technology for developing agent environments in MAS. In Programming
24. Alessandro Ricci, Mirko Viroli, and Andrea Omicini. Give agents their artifacts:
the A&A approach for engineering working environments in MAS. In Proceedings
of the 6th international joint conference on Autonomous agents and multiagent
25. Rafael H Bordini, Jomi Fred Hubner, and Michael J Wooldridge. Programming
26. Shafiq Ur Rehman and Aamer Nadeem. AgentSpeak (I) based testing of au-
tonomous agents. In Software Engineering, Business Continuity, and Education,
27. Christian Anthony L. Go and Won-Hyung Lee. An intelligent belief-desire-
intention agent for digital game-based learning. In Analysis and Design of In-
28. Antonio Barella, Soledad Valero, and Carlos Carrascosa. JGOMAS: new approach
29. A. Barella, C. Carrascosa, and V. Botti. Agent architectures for intelligent virtual
31. Tristan M. Behrens, Koen V. Hindriks, and Jrgen Dix. Towards an environment
2011.
32. Joost Van Oijen, Willem Van Doesburg, and Frank Dignum. Goal-based communica-
tion using bdi agents as virtual humans in training: An ontology driven dialogue
33. Alessandro Ricci, Mirko Viroli, and Andrea Omicini. A general purpose programming
model & technology for developing working environments in MAS. In 5th
International Workshop Programming Multi-Agent Systems (PROMAS 2007), page
34. Thomas Schmickl. How to engineer robotic organisms and swarms? In Bio-Inspired
35. Modular self-reconfigurable robots.
36. Chih-Han Yu and Radhika Nagpal. Sensing-based shape formation on modular
multi-robot systems: a theoretical study. In Proceedings of the 7th international
joint conference on Autonomous agents and multiagent systems-Volume 1, page
7178, 2008.
37. Mark Yim, Paul J. White, Michael Park, and Jimmy Sastra. Modular Self-
Reconfigurable Robots. 2009.
38. M. Park, S. Chitta, A. Teichman, and M. Yim. Automatic configuration recognition
methods in modular robots. The International Journal of Robotics Research, 27(3-
39. Juan Gonzalez-Gomez, Javier Gonzalez-Quijano, Houxiang Zhang, and Mohamed
Abderrahim. Toward the sense of touch in snake modular robots for search and
rescue operations. In Proc. ICRA 2010 Workshop Modular Robots: State of the
40. A modular robotic system with applications to space exploration. Space Mission
Challenges for Information Technology, IEEE International Conference on, 0:8 pp.,
2006.