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 Computación (DSIC)
Camino de Vera s/n, Valencia, Spain
{jrincon,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
[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/

M. Polycarpou et al. (Eds.): HAIS 2014, LNAI 8480, pp. 98–109, 2014.

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[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 fenomenal 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 environments [4,5,6] in so different domains as education [7], entertainment [8,9,10,11], e-commerce[12], health [13,14] and use to VR-based simulations[15].

[5] `http://jacalive.gti-ia.dsic.upv.es/`

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 *JBullet*[6] and *Open Dynamic Engine (ODE)*[7].

Another important feature of any IVE is to offer a high level of graphical realism. Currenly there are in the market some well-developed graphical simulators like *Unity 3D*[8], *Unrealengine UDK*[9] y *Cryengine*[10]. 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)[25,26,27] 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 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

[6] `http://jbullet.advel.cz/`
[7] `http://www.ode.org/`
[8] `http://unity3d.com/unity`
[9] `http://www.unrealengine.com/udk/`
[10] `http://www.crytek.com/cryengine`

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)* [28,29,30]. 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 environments. 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 addressed 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 representing 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 diferenciate between virtual represented 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

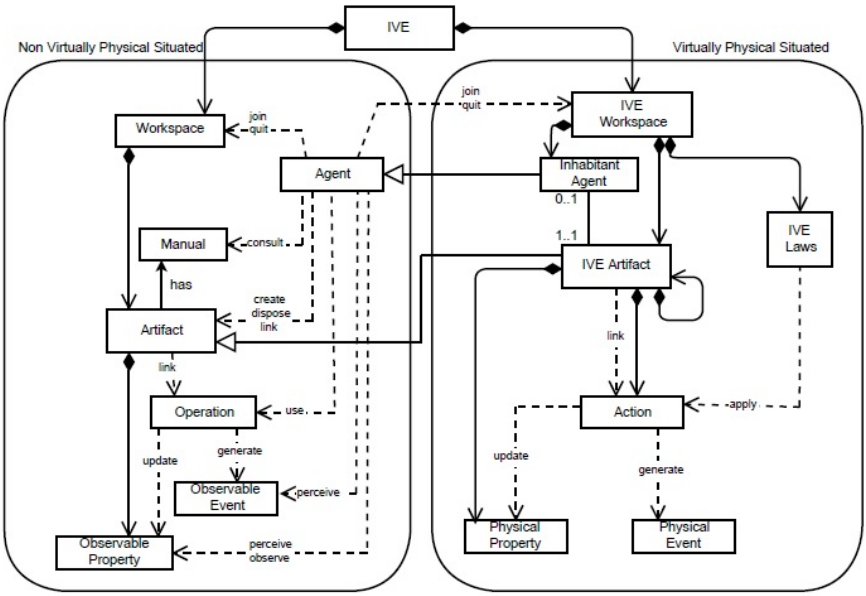

Fig. 1. Meta-modelo MAM5 de un IVE basado en A&A

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 JaCalIVE (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 JaCalIVE 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 JaCalIVE framework.

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

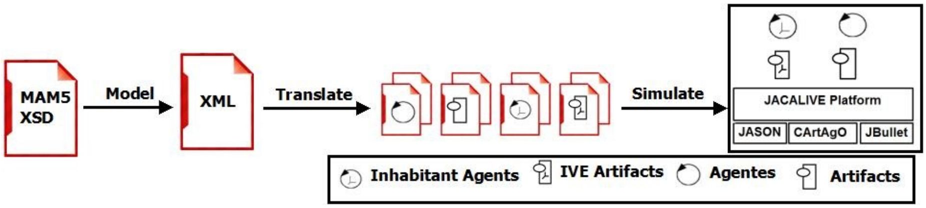

Fig. 2. General Scheme, JaCalIVE

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 [34,35,36] 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 JaCalIVE XSD. The main parts of this XML are: (i) An `IVE_Workspace` called `apodoRobot_Workspaces` as show in Figure 3.

```
<VIRTUAL>
 <IVE_WORKSPACE NAME="apodRobot_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 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.

```xml
<IVE_ARTIFACT NAME="linkedArtifact" LINKEABLE="true">
 <ATTRIBUTES/>
 <PHYSICAL_PROPERTIES>
  <PERCEIVABLE>
    <VECTOR3D NAME = "position">
  <DOUBLE NAME = "x"> 100.0 </DOUBLE>
  <DOUBLE NAME = "y"> 80.0 </DOUBLE>
  <DOUBLE NAME = "z"> 0.0 </DOUBLE>
 </VECTOR3D>
 <VECTOR3D NAME = "velocity">
  <DOUBLE NAME = "x"> 1.0 </DOUBLE>
  <DOUBLE NAME = "y"> 1.0 </DOUBLE>
  <DOUBLE NAME = "z"> 1.0 </DOUBLE>
 </VECTOR3D>
 <VECTOR3D NAME = "orientation">
  <DOUBLE NAME = "x"> 1.0 </DOUBLE>
  <DOUBLE NAME = "y"> 0.0 </DOUBLE>
  <DOUBLE NAME = "z"> 0.0 </DOUBLE>
 </VECTOR3D>
    <VECTOR3D NAME = "joint">
  <DOUBLE NAME = "x"> 0.0 </DOUBLE>
  <DOUBLE NAME = "y"> 0.0 </DOUBLE>
  <DOUBLE NAME = "z"> 0.0 </DOUBLE>
```

Fig. 4. XML that configures the IVE_Artifacts

```xml
<INHABITANT_AGENT NAME="Robot">
 <ATTRIBUTES/>
 <BODY_ARTIFACT>
  <ITEM ID="0"/>
  <ITEM ID="1"/>
 </BODY_ARTIFACT>
 <FILE NAME="apodRobotJason.asl"/>
</INHABITANT_AGENT>...

<IVE_LAW NAME="Gravity">
 <VECTOR3D NAME="gravity">
  <DOUBLE NAME="x"> 0.0 </DOUBLE>
  <DOUBLE NAME="y"> -9.8 </DOUBLE>
  <DOUBLE NAME="z"> 0.0 </DOUBLE>...
 </VECTOR3D>
 <ACTIONS>
  <ITEM NAME="move"/>
 </ACTIONS>
</IVE_LAW>
```

Fig. 5. XML that configures the Inhabitant Agents

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 Unity 3D. Figure 6 shows an snapshot excerpt of the simulation.

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

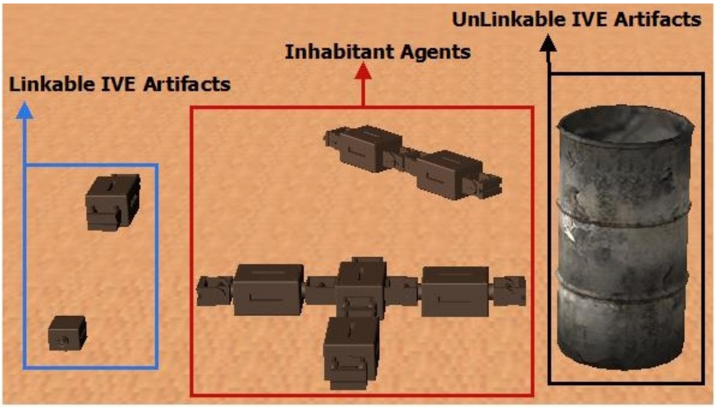

Fig. 6. Excerpt of the case study simulation using Unity 3D

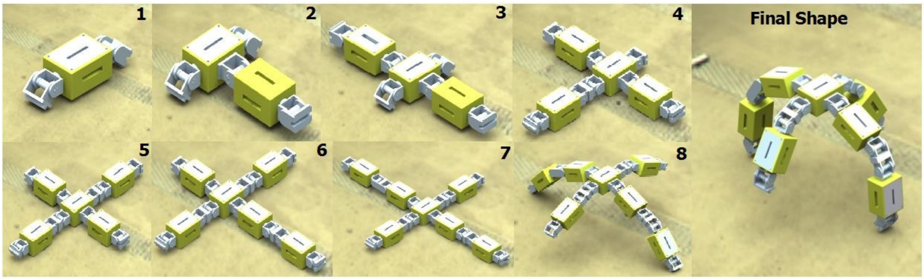

Fig. 7. Modular robotic sequence

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.

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 JaCalIVE framework can be easily modified thanks to the XML modellation 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.

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

1. Aylett, R., Luck, M.: Applying artificial intelligence to virtual reality: Intelligent virtual environments. Applied Artificial Intelligence 14, 3–32 (2000)
2. Barella, A., Ricci, A., Boissier, O., Carrascosa, C.: MAM5: Multi-Agent Model For Intelligent Virtual Environments. In: 10th European Workshop on Multi-Agent Systems (EUMAS 2012), pp. 16–30 (2012)
3. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems 17(3), 432–456 (2008)
4. Yu, C.-H., Nagpal, R.: Distributed consensus and self-adapting modular robots. In: IROS 2008 Workshop on Self-Reconfigurable Robots and Applications (2008)
5. Lidoris, G., Buss, M.: A multi-agent system architecture for modular robotic mobility aids. In: European Robotics Symposium 2006, pp. 15–26 (2006)
6. Yu, C.-H., Nagpal, R.: A self-adaptive framework for modular robots in a dynamic environment: theory and applications. The International Journal of Robotics Research 30(8), 1015–1036 (2011)
7. Barbero, A., González-Rodríguez, M.S., de Lara, J., Alfonseca, M.: Multi-agent simulation of an educational collaborative web system. In: European Simulation and Modelling Conference (2007)
8. Ranathunga, S., Cranefield, S., Purvis, M.K.: Interfacing a cognitive agent platform with a virtual world: a case study using second life. In: AAMAS, pp. 1181–1182 (2011)
9. Andreoli, R., De Chiara, R., Erra, U., Scarano, V.: Interactive 3d environments by using videogame engines. In: Proceedings of the Ninth International Conference on Information Visualisation, pp. 515–520. IEEE (2005)
10. Roncancio, C., Gómez G-B, J., Zalama, E.: Modeling virtual agent behavior in a computer game to be used in a real enviroment. In: Demazeau, Y., et al. (eds.) Trends in PAAMS. AISC, vol. 71, pp. 623–630. Springer, Heidelberg (2010)
11. Dignum, F.: Agents for games and simulations. Autonomous Agents and Multi-Agent Systems 24(2), 217–220 (2011)
12. dos Santos, C.T., Osorio, F.S.: AdapTIVE: an intelligent virtual environment and its application in e-commerce. In: Proceedings of the 28th Annual International Computer Software and Applications Conference, COMPSAC 2004, vol. 1, pp. 468–473 (September 2004)
13. Tian, J., Tianfield, H.: A multi-agent approach to the design of an E-medicine system. In: Schillo, M., Klusch, M., Müller, J., Tianfield, H. (eds.) MATES 2003. LNCS (LNAI), vol. 2831, pp. 85–94. Springer, Heidelberg (2003)

14. Mago, V.K., Syamala Devi, M.: A multi-agent medical system for indian rural infant and child care. In: IJCAI, pp. 1396–1401 (2007)
15. Anastassakis, G., Ritchings, T., Panayiotopoulos, T.: Multi-agent systems as intelligent virtual environments. In: Baader, F., Brewka, G., Eiter, T. (eds.) KI 2001. LNCS (LNAI), vol. 2174, pp. 381–395. Springer, Heidelberg (2001)
16. Garcá-Pardo, J.A., Carrascosa, C.: Social Welfare for Automatic Innovation. In: Klügl, F., Ossowski, S. (eds.) MATES 2011. LNCS, vol. 6973, pp. 29–40. Springer, Heidelberg (2011)
17. Kazemi, A., Fazel Zarandi, M.H., Moattar Husseini, S.M.: A multi-agent system to solve the production-distribution planning problem for a supply chain: a genetic algorithm approach. Int. J. Adv. Manuf. Technol. 44(1-2), 180–193 (2009)
18. Dimuro, G.P., da Rocha Costa, A.C., Gonçalves, L.V., Hubner, A.: Interval-valued hidden markov models for recognizing personality traits in social exchanges in open multiagent systems (2008)
19. Orkin, J., Roy, D.: Semi-automated dialogue act classification for situated social agents in games. In: Dignum, F. (ed.) Agents for Games and Simulations II. LNCS (LNAI), vol. 6525, pp. 148–162. Springer, Heidelberg (2011)
20. Woźniak, M., Graña, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. Information Fusion 16, 3–17 (2014)
21. Jia, L., Zhenjiang, M.: Entertainment oriented intelligent virtual environment with agent and neural networks. In: IEEE International Workshop on Haptic, Audio and Visual Environments and Games, HAVE 2007, pp. 90–95 (October 2007)
22. Corchado, E., Wonźiak, M., Abraham, A., de Carvalho, A.C.P.L.F., Snásel, V.: Recent trends in intelligent data analysis. Neurocomputing 126, 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)
23. Ricci, A., Viroli, M., Omicini, A.: The A&A programming model and technology for developing agent environments in MAS. In: Dastani, M., El Fallah Seghrouchni, A., Ricci, A., Winikoff, M. (eds.) ProMAS 2007. LNCS (LNAI), vol. 4908, pp. 89–106. Springer, Heidelberg (2008)
24. Ricci, A., Viroli, M., Omicini, A.: 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 Systems, p. 150 (2007)
25. Bordini, R.H., Hübner, J.F., Wooldridge, M.J.: Programming multi-agent systems in AgentSpeak using Jason. J. Wiley, Chichester (2007)
26. Rehman, S.U., Nadeem, A.: AgentSpeak (L) based testing of autonomous agents. In: Kim, T.-H., Adeli, H., Kim, H.-K., Kang, H.-J., Kim, K.J., Kiumi, A., Kang, B.-H. (eds.) ASEA 2011. CCIS, vol. 257, pp. 11–20. Springer, Heidelberg (2011)
27. Go, C.A.L., Lee, W.-H.: An intelligent belief-desire-intention agent for digital game-based learning. In: Analysis and Design of Intelligent Systems Using Soft Computing Techniques, pp. 677–685. Springer (2007)
28. Barella, A., Valero, S., Carrascosa, C.: JGOMAS: new approach to AI teaching. IEEE Transactions on Education 52(2), 228–235 (2009)
29. Barella, A., Carrascosa, C., Botti, V.: Agent architectures for intelligent virtual environments, pp. 532–535. IEEE (November 2007)
30. Barella, A., Carrascosa, C., Botti, V., Martí, M.: Multi-agent systems applied to virtual environments: A case study. In: Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology, pp. 237–238. ACM (2007)

31. Behrens, T.M., Hindriks, K.V., Dix, J.: Towards an environment interface standard for agent platforms. Ann. Math. Artif. Intell. 61(4), 261–295 (2011)
32. van Oijen, J., van Doesburg, W., Dignum, F.: Goal-based communication using BDI agents as virtual humans in training: An ontology driven dialogue system. In: Dignum, F. (ed.) Agents for Games and Simulations II. LNCS, vol. 6525, pp. 38–52. Springer, Heidelberg (2011)
33. Ricci, A., Viroli, M., Omicini, A.: A general purpose programming model & technology for developing working environments in MAS. In: 5th International Workshop "Programming Multi-Agent System" (PROMAS 2007), pp. 54–69 (2007)
34. Schmickl, T.: How to engineer robotic organisms and swarms? In: Meng, Y., Jin, Y. (eds.) Bio-Inspired Self-Organizing Robotic Systems. SCI, vol. 355, pp. 25–52. Springer, Heidelberg (2011)
35. Modular self-reconfigurable robots
36. Yu, C.-H., Nagpal, R.: 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, vol. 1, pp. 71–78 (2008)
37. Yim, M., White, P.J., Park, M., Sastra, J.: Modular Self-Reconfigurable Robots (2009)
38. Park, M., Chitta, S., Teichman, A., Yim, M.: Automatic configuration recognition methods in modular robots. The International Journal of Robotics Research 27(3-4), 403–421 (2008)
39. Gonzalez-Gomez, J., Gonzalez-Quijano, J., Zhang, H., Abderrahim, M.: Toward the sense of touch in snake modular robots for search and rescue operations. In: Proc. ICRA 2010 Workshop on Modular Robots: State of the Art, pp. 63–68 (2010)
40. A modular robotic system with applications to space exploration. In: IEEE International Conference on Space Mission Challenges for Information Technology, 0:8 p. (2006)