An Integrated Architecture for Multiagent Virtual Worlds for Performing Adaptive Testing Games

Sandeep Virwaney Fuhua Lin Grant McClure Robert Heller
Athabasca University, Canada
oscarl@athabascau.ca

***Abstract* — One of the key success factors that contribute towards the creation and sustenance of online (2D and 3D) virtual worlds for learning might be to provide game-style educational activities. However, there is no development platform available which can meet the inherent system requirements including usability of platform and scalability to modern massively multiplayer online games, yet focused on engaging learning for the individual user. Work has been done with software agents in the context of multi-agent systems (MAS), and it makes sense to try to leverage that work when it comes to modeling functional modules, controlling realistic non-player characters (NPCs), and Personal Assistants for Learning (PALs) in a virtual learning world. There are challenges to integrating a multi-agent system into a virtual world including concerns with synchronization, communication, monitoring, efficiency, and control. This paper describes the design and implementation of an integrated architecture for performing and facilitating quiz games for adaptive testing with a multi-agent system JADE/Jason and a 3D virtual world engine Open Wonderland.**

Keywords:* *virtual learning environments; multiagent systems

I. INTRODUCTION

Educational theories classify learning activities as constructivist, constructionist, or situated [14]. The three types of learning are well served by virtual worlds due to its ability to mediate world exploration and construction, to map a user to any character of choice, and to provide shared virtual three dimensional space. Through its interactive environment, repetition, and one-to-one experimentation, virtual worlds can help improve knowledge retention and student motivation. By simulating chemistry or physics experiments, knowledge can be transferred without increased safety risks associated with real laboratories. Through the Internet, virtual worlds can facilitate distance learning in places where schools or specialized teachers are scarce, and so on.

There have been relatively mature technologies for physical modeling for virtual objects, modeling virtual humans or non-player characters (NPCs) in virtual worlds in their appearance, gestures, kinematics, and physical properties. Avatars are controlled by the user during interaction. Like objects, virtual humans or NPCs are controlled by a simple script.

To build a smart virtual world, personalized and adaptive virtual learning environment, it is desirable to incorporate intelligence to virtual worlds. Work has been done with software agents in the context of multi-agent systems, and it makes sense to try to leverage that work when it comes to modeling functional modules for pedagogical purposes, controlling realistic non-player characters (NPCs), and maintaining Personal Assistants for Learning (PALs) in a virtual learning environment.

The research is interdisciplinary in nature. The domains involved include human-computer interaction, psychology, and education. The problem is that there are no agent development tools that are intended to be easy to use by non-programmers. To develop such tools we need to have a novel integrated system architecture that can incorporate Artificial Intelligence (AI) technologies such as machine learning, data mining, natural language processing, affective computing, and models and algorithms for building the agents.

A. MAS in Virtual Worlds

Multi-agent systems (MAS) have been adopted by researchers in virtual worlds. MAS is particularly well-suited to application domains where virtual entities (agents) are self-directed and can actively pursue their goals within an environment that they can interact with, including interactions with other entities that are also in pursuit of their own goals. It is easy to see how they are ideally suited for modeling people — they are active and social in a way similar to people.

There are two perspectives to consider when applying MAS into virtual worlds. One is from the artificial life (ALife) community while the other is from classical AI community. In the ALife community, the term complex dynamical systems is usually preferred, because it also includes physical inorganic systems, where the individual agents or components, such as molecules or sand grains, only have limited agent characteristics. ALife has strong roots in biology rather than psychology, and focuses on the *emergence* of behavior in large populations of agents [1].

In the AI community, an agent is assumed to have certain elementary sensory-motor abilities, so that it can perceive aspects of the environment and, depending on this information and its own state, perform certain behaviors. Rather than looking for emergence, as is common in the field of ALife, MAS is usually employed to solve some particular problems or achieve particular tasks, such as path-finding, coordinated control [2]. The individual agents are endowed with centralized control similar to that employed in the classical approach.

B. QuizMASter

In classrooms, teachers usually use quiz games to create some interesting activities. The purposes of quiz games for the classes include:

(1) reviewing and reinforcing previously taught material,

(2) warming up or ending lesson with a socially engaging activity,

(3) encapsulating the basic unit of conversation.

We have been designing and developing engaging pedagogical agents to enhance the engagement of students in a game-based learning environment QuizMASter [4-6], using a multi-agent system (MAS) approach. It helps students undergo adaptive testing and collaborative learning through friendly competition. Conceptually, QuizMASter is designed to be similar to a TV game show, where a small group of contestants compete by answering questions presented by the game show host. A prototype of QuizMASter using a JADE MAS was successfully implemented on Open Wonderland [7]. The implementation enabled an opportunity to explore the affective dimension of gaming situation by controlling the affective channels of information between avatars. Specifically, the QuizMASter agents can be enhanced to operate within the dimensions of Transformed Social Interaction (TSI) [8] to maintain high levels of engagement. Bailenson et al. (2008) proposed the theory of Transformed Social Interaction to describe the transformation of interaction in virtual environments. According to the theory, real-time transformations can be classified into three categories or dimensions: self-representation, sensory-abilities and situational context [8]. These transformations empower avatars to complement human perceptual abilities [12]. For the autonomous agent and the avatar it represents, the ability to process eye gaze or motion data is advantageous in any implementation of virtual world applications. Situational context deals with transformations that alter the spatial or temporal structure of a conversation. For example, the communication and positions between agents and students can be optimally configured in terms of the geographical setup of a classroom. A class of 20 students can sit directly in front of a virtual instructor, and perceive the rest of the students as sitting farther away. Furthermore, by altering the flow of rendered time in the communication session, users can implement strategic uses of rewind and fast forward during a real-time interaction to increase comprehension and efficiency.

This paper focuses on the roles of agents in multi-agent virtual learning environments and some architectural designs we have proposed and developed for multi-agent virtual learning environments. We investigate the effectiveness of the use of agents to facilitate virtual world based quiz games.

II. RELATED WORK

Two different applications towards multi-agent virtual learning environments are surfacing: virtual classrooms and fantasy worlds. Virtual classrooms have the look and feel of a regular classroom and are often made to look like a replica of the sponsoring school. Second Life (SL) is often the choice for building the 3D virtual classrooms. The results are a mixed bag with some reporting that SL has too many difficulties and is not very game-like as such classrooms lack game scenarios. Fantasy worlds teleports the learner to a different place or time. For example, Virtual Singapura (VS) [15], an example of a fantasy world takes the learner to nineteen century Singapore in the throes of disease epidemic. The VS project is one of the first to integrate an intelligent agent architecture with a virtual world to explore ways in which adaptive synthetic characters might enhance the learner's experience in a virtual world and perhaps to enhance learning as well [15].

There has been some work done over the last decade when it comes to creating agent control mechanisms for 3D virtual environments, for example, Gamebots [9] and Pogamut [10]. Dingum et al. (2008) provide an excellent overview of the state of the art when it comes to incorporating MAS concepts with gaming systems or other 3D virtual environments [11]. According to the literature, it looks like most work to date has relied on the Unreal Tournament engine and Gamebots. There are a number of positives to this technology including the robustness of the environment. The downsides include the fact that it is not open source, and there are therefore limitations in what can be done in terms of ensuring agents have all the information they need and that synchronization works perfectly. There are also limitations in communications that means that agents aren’t able to take full advantage of the social actions in a MAS environment.

III. THE ROLES OF AGENTS

Agents are encapsulated computer systems that are designed to behave flexibly and somewhat autonomously to achieve some goal(s). Agents are situated in some environment and have some autonomy and capabilities to observe that environment. They can communicate those observations to other agents. This makes them particularly suited for distributed virtual learning environments. Such environments are open in nature as either students or teachers can enter or leave the virtual world at will.

They may be either visible or invisible. We classify the agents in a multi-agent virtual learning environment into the following types:

A. Game agents

First, the game agents are designed to realize the game functions. The game agents include game management agents, game control agents, user management agents, and domain knowledge management agents. Gaming frameworks are quite mature when it comes to optimizing graphics and physics—current games are visually stunning. There is more to a gaming experience than what a player sees. How a game is structured and the kind of narrative it plays also lends itself to the credibility of the virtual world a player interacts with. This includes the behavior of NPCs in the environment. If an NPC does not act in a believable way it detracts from the realism of the player’s game experience.

B. Pedagogical agents.

Pedagogical agents are agents that could provide pedagogy to the games, including Instructor agents that work with the Host Agent to ensure the subject, difficulty, and content of the quiz are appropriate for each game; Behavior agents that during each game collects information, such as response time and score needed for PAL to update the student model.

C. PALs

Personal agents or Personal Assistants for Learning (PALs) are designed for analyzing the user's profile, learning attributes, and competencies keeping track of individual interests, preferences, motivations, and goals of the human participants (i.e., learners) and building student models.

D. NPCs

To make the game-based learning environment more realistic and immersive, a MAS is ideal for modeling virtual students and a virtual audience which are implemented as NPCs or bots. Many researchers in the virtual worlds community have noted that city replicas, malls, bars, or educational places often look interesting but abandoned at the same time. This socially isolating aspect could be mitigated by deploying MAS controlled NPCs who could inhabit virtual worlds to create impressions of the societies we live in. Also, they could be used to substitute the work of real people by fulfilling different tasks such as demonstrating objects, explaining manuals or guiding the way to new locations.

AI techniques have been used to drive NPCs, but this is usually done in a prescriptive manner that doesn't give a lot of freedom to NPC actions. These NPCs require simple scripting as the tasks they are following are simple procedures and need almost no interaction with the environment. Nevertheless, NPCs enhance the immersive feeling in a virtual world and provide answers to users that are seldom possible by other means. Moreover, we can increase the intelligence of the NPCs through implementing MAS *communication* and *coordination mechanisms* and further sensory feedback to actually enable interaction between human avatars and bots that are representing people.

Finally, the communication and coordination among these three types of agents would be extremely important to make the learning environment smart. To enable game-based learning, it is crucial for the pedagogical agents to make the quizzes interactive, interesting, appropriate in the degree of difficulty, and adaptive in topics given a group of students with different student models. Based on the user models, the game agents will also be able to make decisions intelligently, e.g. to optimize the grouping and seating/standing positions of participants according to their preferences.

IV. IMPLEMENTATION

The primary goal of this research is to construct an integrated intelligent virtual world for performing and facilitating quiz game-style adaptive testing. For a 3D VW, Open Wonderland [7] is used. For a MAS, a combination of JADE and Jason is used, leveraging the strengths of both where appropriate. As such, the framework needs to have interfaces to both. We have proposed an architecture, the WL-MAS interface, that is the foundation on which the educational game, QuizMASter, has been built. The WL-MAS interface are used by agents controlling NPCs as well as agents controlling other aspects of the game, such as score keeping and user modeling.

The following work has been completed to ascertain the technical feasibility and limitations creating agent-controlled NPCs in a 3D virtual world like Open Wonderland.

A. JADE-OWL Integration

As an initial step, we integrated Open Wonderland and JADE by starting a JADE server separate from Open Wonderland [12]. JADE (http://jade.tilab.com/) is a middleware that facilitates the development of multiagent system. JADE agents can be started inside the Open Wonderland world by simply using *jade.core.Runtime* and which enables the agent to send pertinent information back to OWL in the form of messages to a TCP port. The interface is a modified OWL module to which code has been added to start a JADE agent. The JADE agent is started via a runtime call. Therefore, the agent will be started on a local computer. However, the rest of the agents that make up the MAS can be on other computers. The agents started by the OWL module will communicate with the rest of the JADE agents using the FIPA specified protocol. One of the agents in the MAS communicates to a TCP port that has been activated on another or the same OWL module.

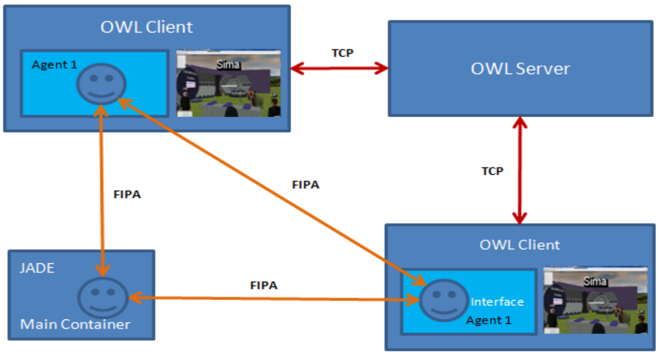


Figure 1: JADE-OWL integration.

JADE followed the FIPA assumption that only the external behavior of system components should be specified, while leaving the implementation details and internal architectures to agent developers. It implements a very general agent model that can be easily specialized to realize both reactive and Belief, Desires, and Intentions (BDI) architectures. Moreover, the behavior abstraction of our agent model allows simple integration of external software into one of the agent tasks.

B. Jason/JADE-OWL Integration

The results of our previous work showed the possibilities and challenges that arise when the systems aren't coupled enough [12]. These challenges informed our decision to simplify and integrate everything right into Open Wonderland. Therefore, our second model added a Jason module to the Open Wonderland server that takes an NPC and controls its movement in the environment (see Figure 2). Jason (http://jason.sourceforge.net/Jason/Jason.html) is an interpreter for an extended version of AgentSpeak and a platform for the development of multiagent systems.

The NPC runs on the Open Wonderland server and pulls an AgentSpeak() asl file from the file system (in the Open Wonderland server cache), sets up its belief system and goals, and then runs. As yet it doesn't get any perceptual information from the environment, but is able to guide the direction of the NPC. It will be easy to extend this to have the NPC make any available action that an avatar can do within Open Wonderland. The percept feedback to the Jason agent still needs to be completed. The MAS and the Open Wonderland will be more tightly coupled than what had originally been planned so that development, deployment, and management of QuizMASter are easier when handled within a single environment. The Open Wonderland module system makes it simple to extend Open Wonderland to include the framework and any other functionality that is required [7]. Figure 3 shows the screen shot of Jason and Open Wonderland integration.

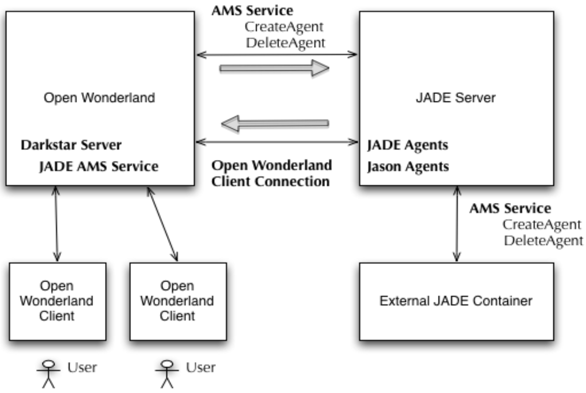


Figure 2: Jason/JADE-OWL integration.

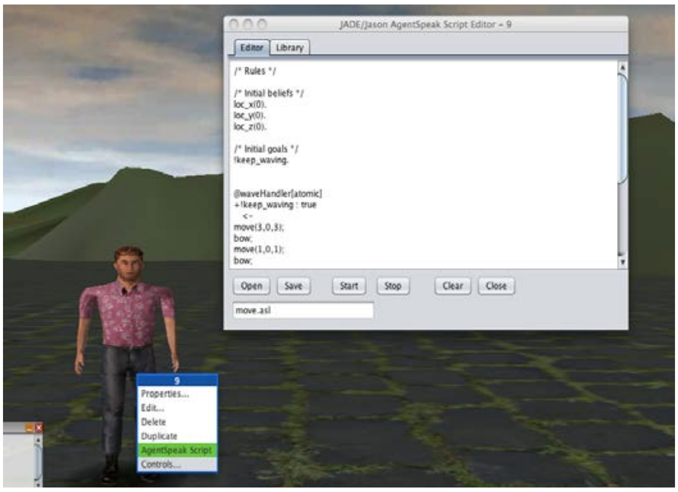


Figure 3: Jason and Open Wonderland integration.

C. *Jason-CArtAgO-OWL Architecture*

Our third model is CArtAgO (Common ARTifact infrastructure for AGents Open environments) [4]. CArtAgO is a general purpose framework that makes it possible to program and execute virtual environments for multi-agent systems. CArtAgO is based on the Agents & Artifacts (A&A) meta-model for modeling and designing MASs. Since these artifacts can be modeled in Java, some of the core components of the Open Wonderland. In the current implementation as a prototype, some agents are temporally implemented as artifacts. A MaryTTS server is added to the system to realize speech synthesis function with Open MARY TTS which is an open-source, multilingual Text-to-Speech Synthesis platform written in Java (http://mary.dfki.de/).

One of the main advantages of leveraging the capabilities of a Jason-CArtAgO MAS is that it enables customization of the rendering of visible artifacts for different clients. This advantage was crucial for implementing the three principles of TSI theory [8]. In particular, the principle of self-representation in NPC creation is described below.

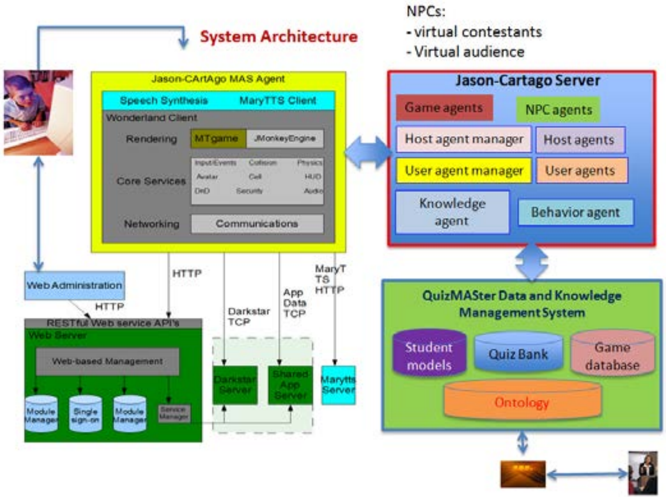


Figure 4: Jason-CArtAgO-OWL architecture.

1) *NPC Creation and Facial Morphing*

Evolver (http://evolver.com) was an online tool that enables one to model, texture, and rig one's own 3D character in minutes. It has the capability to design the face of an NPC using a 2D photograph and further customize the body type, eyes, skin-complexion and wearable costumes of the avatar being developed. This capability of *Evolver* was used to create an NPC of one of the contestants, the host agent for player Oscar and player Steve as show in Figure 5 below. As shown by Bailenson et al. (2008), when facial images incorporate personal features, user satisfaction ratings increase.

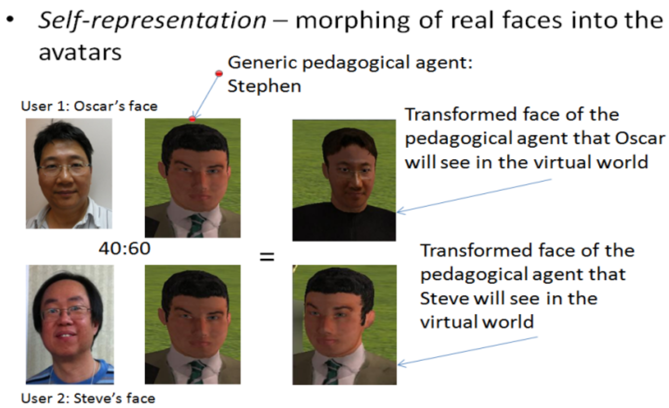


Figure 5: Creating the face of the host agent of players.

Using the evolver-multimesh module, all NPCs were imported into the Wonderland virtual environment. Furthermore, using the scripting-component module and the NPC module, these NPCs were given capability to produce animations such as clapping, cheering and public speaking. In CArtAgo model, NPCs have been wrapped as artifacts in order for the Jason agents to manipulate, manage, and communicate with them. They can converse with contestants and agents that could react to external events such as contestant answers to questions, calculate and compute scores amongst other things. Figure 6 shows the audience performing the cheering and clapping animation.

Figure 6: The virtual audience performing the cheering and clapping animation.

Figure 7 shows the **Host agent** performing public speaking animation. By synchronizing the cheering and clapping animation with appropriate sound effects, and also host public speaking animation of the host with synthesized speech delivered by MaryTTS, an attempt was made to make these gestures seem as real as possible to the contestant.

Figure 7: the host agent performing public speaking animation.

2) Agent interactions

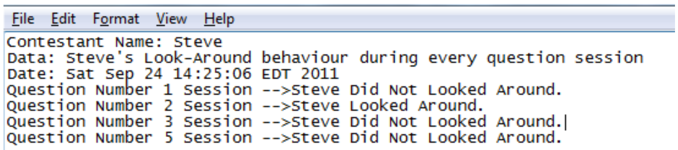
First the **Game agent** starts the Open Wonderland client. In order to achieve this goal, its first plan is to create an artifact called *WLStarterArtifact*. Then, a focus is done on the *WLStarterArtifact* so that any signal generated by *WLStarterArtifact* is observable by all agents in the current default workspace.

The next goal of the **Game agent** is to invoke the main entry point of the wonderland client which is *JmeClientMain*. Once the invocation of the wonderland client is successful, the *WLStarterArtifact* sends a *readytostart* signal which is perceivable by all agents within the default workspace. As soon as the **Host agent** percepts *readytostart* signal, it starts executing the plan with *QMIntroArtifact*. It goes on to execute the plan of *doIntroductionStage*.

Once the Introduction stage is done, the **Host agent** sends a signal "*introduce"* which is perceivable by **the Game agent**. After perceiving the "*introduce"* signal, the **Game agent** executes the plan with *ClockArtifact* – which starts and stops the timer for every question, and communicates with the **Host agent** to carry out the entire question and answer session, including displaying the questions, making sure that the **Host agent** delivers the question verbally along with necessary comments to the contestant. The Host agent performs this by communicating with the **Instructor agent and** *ClockArtifact*. Figure 9 shows the result of this stage in the form of a screen captured image.

Figure 9: A player is answering the question with a mouse.

At the beginning of the QuizMASter show during the initialization stage, contestants were given control of their avatars by enabling them to change the look direction using the left and the right arrow keys. When the show starts, during each question and answer sessions, the movements of the user are monitored by **the Behavior agent** using *CellTransform* which holds the translation and rotation parameters of each cell in Open Wonderland environment. All contestant movements are recorded in a log file in the name of the contestant. If it is detected that the contestant is looking away from the **Host agent**, the contestant cell is transformed using *CellTransform* so that he/she is facing the direction of the Host agent and the **Host agent** requests the user to pay attention to the quiz. This is how the 2nd dimension of TSI is implemented in QuizMASter. Figure 10 shows an example of the log files, which is generated for contestant named Steve.



File Edit Format View Help

```
Contestant Name: Steve
Data: Steve's Look-Around behaviour during every question session
Date: Sat Sep 24 14:25:06 EDT 2011
Question Number 1 Session -->Steve Did Not Looked Around.
Question Number 2 Session -->Steve Looked Around.
Question Number 3 Session -->Steve Did Not Looked Around.|
Question Number 5 Session -->Steve Did Not Looked Around.
```

Figure 10: Contestant Steve's log file of his look_around behavior during Every Question Answer Session.

We successfully tested the prototype with an online course ENLI 255 of Athabasca University and two real players located in different places in Canada. The OWL server is installed in Northern Alberta of Institute of Technology of Canada.

V. CONCLUSIONS

This paper introduces an architectural design and prototype implementation of adaptive testing games within a virtual world.

In order to retain the functionality of having a MAS within the QuizMASter framework that was previously developed [12], we enhanced the architecture by incorporating a more versatile BDI agent model with AgentSpeak using Jason into it. Furthermore, there was an opportunity to implement the Agent and Artifact model proposed by CartAgO tool, that fully supports agents implemented using Jason and provides a bridge for agents developed using Jason. By leveraging the power of CartAgO and being able to drive and control the entire Open Wonderland client using Jason and CartAgO, QuizMASter could be developed primarily using artifacts whereby it can be driven entirely by intelligent agents. The seemingly endless possibilities of using QuizMASter as a test-bed for conducting current and future experiments using 3D virtual world along with sophisticated intelligent agents will provide a valuable research tool to investigate game based learning and adaptive assessment.

The main advantage of Jason-CArtAgO MAS is that it bridges the gap between designing MAS and programming MAS.

Thus far, we have identified the roles of agents and the three different architectures we developed to investigate MAS in a game environment. To make environments more effective for educational applications, we need to provide a personalized and customized environment where students feel encouraged. We are working on user modeling for such environments but more work about the coordination mechanisms for dynamic game formation and adaptive quizzes sequencing, and incorporation of the intelligence of the agents with the architecture proposed needs to be done.

To meet the needs of the next generation learners, educators must use innovative methodologies and approaches for solving problems in collaborative, disparate environments in a manner that improves learning outcomes. We will continue to explore the roles of agents and agent technologies in facilitating and supporting assessment-based learning wherein learners engage in self-directed learning and are able to test out of education or training by successfully completing validated assessments outside of structured and formal learning instances.

ACKNOWLEDGEMENT

We thank NSERC discovery grant of Canada and Canada Foundation for Innovation (CFI), Athabasca University, and communities for Open Wonderland, Jason, and JADE for their support and help to this research.

REFERENCES

[1] Pfeifer, R. and J. Bongard, (2007). How the body shapes the way we think: a new view of intelligence, MIT Press.

[2] Weiss, G. (2000). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press.

[3] Avlett, R., M. Vala, P. Sequeira, A. Paiva, (2007). FearNot! – An Emergent Narrative Approach to Virtual Dramas for Anti-bullying Education, Virtual Storytelling. Using Virtual Reality Technologies for Storytelling, LNCS 4871/2007, 202-205

[4] Ricci, A., M. Viroli, and A. Omicini. (2007). 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 Berlin / Heidelberg.

[5] Dutchuk, M., K. A. Muhammadi, & F. Lin **(**2009**),** QuizMASter - A Multi-Agent Game-Style Learning Activity, EduTainment 2009, Aug 2009, Banff, Canada, Learning by Doing, (eds.), M. Chang, R. Kuo, Kinshuk, G-D Chen, M. Hirose, LNCS 5670, 263-272.

[6] Weng, M., F. Lin, T. K. Shih, M. Chang, I. Fakinlede, (2011). A Conceptual Design of Multi-Agent based Personalized Quiz Game, The 11th IEEE International Conference on Advanced Learning Technologies (ICALT), Athens, Georgia, USA

[7] Kaplan, J. and Yankelovich, N. (2011). Open Wonderland: An Extensible Virtual World Architecture, *IEEE Internet Computing*, **15**(5), 38 - 45.

[8] Bailenson, J. N., Yee, N., Blascovich, J. & Guadagno, R.E. (2008). Transformed Social Interaction in Mediated Interpersonal Communication. In Elly A. Konijn, Sonja Utz, Martin Tanis & Susan B. Barnes (Eds.), *Mediated Interpersonal Communication*. 77–99. New York and London: Routledge.

[9] Kaminka, G. A., Veloso, M. M., Schaffer, S., Sollitto, C., Adobbati, R., Marshall, A. N., et al. (2002). GameBots: a flexible test bed for multiagent team research. Communications of the ACM, 45(1).

[10] Gemrot, J., Kadlec, R., Bída, M., Burkert, O., Píbil, R., Havlíček, J., et al. (2009). Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents, 1–15.

[11] Dignum, F., Westra, J., van Doesburg, W. A., & Harbers, M. (2009). Games and Agents: Designing Intelligent Gameplay. International Journal of Computer Games Technology, 1-19.

[12] Blair, J. and F. Lin, (2011). An Approach for Integrating 3D Virtual Worlds with Multiagent Systems, ISeRim workshop - IANA 2011, Singapore.

[13] Thalmann, D., S. Musse, and M. Kallmann, (2000), From Individual Human Agents to Crowds, Informatique, 2000(No. 1), 6-11.

[14] Moshell, M. and C. Hughes (2002), Virtual Environments as a Tool for Academic Learning, in K. Stanney (Ed.), The Handbook of Virtual Environments Technology, Erlbaum, Mahwah, NJ, 893-910.

[15] Michael J. Jacobson, Beaumie Kim, Chunyan Miao, Zhiqi Shen, Mark Chavez, *Design Perspectives for Learning in Virtual Worlds*, in Jacobson, M. J., & Reimann, P. (Eds.). (2009). Designs for learning environments of the future: International perspectives from the learning sciences. Springer, 111-141.