

# Simulating Interactive Learning Scenarios with Intelligent Pedagogical Agents in a Virtual World through BDI-Based Agents

<http://dx.doi.org/10.3991/ijep.v3i2.2456>

Mohamed Soliman<sup>1</sup> and Christian Guetl<sup>1,2</sup>

<sup>1</sup>Graz University of Technology, Graz, Austria

<sup>2</sup>Curtin University of Technology, Perth, Western Australia

**Abstract**—Intelligent Pedagogical Agents (IPAs) are designed for pedagogical purposes to support learning in 3D virtual learning environments. Several benefits of IPAs have been found adding to support learning effectiveness. Pedagogical agents can be thought of as a central point of interaction between the learner and the learning environment. And hence, the intelligent behavior and functional richness of pedagogical agents have the potential to reward back into increased engagement and learning effectiveness. However, the realization of those agents remains to be a challenge based on intelligent agents in virtual worlds. This paper reports the challenging reasons and most importantly an approach for simplification. A simulation based on BDI agents is introduced opening the road for several extensions and experimentation before implementation of IPAs in a virtual world can take place. The simulation provides a proof-of-concept based on three intelligent agents to represent an IPA, a learner, and learning object implemented in JACK and Jadex intelligent agent platforms. To that end, the paper exhibits the difficulties, resolutions, and decisions made when designing and implementing the learning scenario in both domains of the virtual world and the agent-based simulation while comparing the two agent platforms.

**Index Terms**—Intelligent Agents Simulation; Intelligent Pedagogical Agents; Immersive Virtual Learning Environments;

## I. BACKGROUND AND MOTIVATION

Learners in a virtual world enjoy rich visualization, collaboration, and continuous remote availability of learning resources in the environment. Thus virtual worlds were considered for adoption for several fields in e-education. One of the significant fields is online labs and remote experimentation that add value to online engineering education given the scalability, convenience, availability, and cost effectiveness. And therefore, several research efforts attempted to employ virtual worlds for those fields [1][2][3].

In a virtual world, distant users are immersed into the 3D virtual environment being represented by avatars. Communication among the different users is possible through text chat and voice given the immersive 3D scene providing visual collaboration opportunities. This visual characteristic enables simulations [2], or visualization of a remote lab experimentation [1], to be added for potential learners to interact with providing a new learning setting. Virtual worlds thus introduced authentic learning experi-

ences through rich 3D visualizations combined will collaboration abilities. Generally, in those settings, learning support is assumed through collaborating avatars that can represent a human tutor for example.

While virtual worlds provide appealing feature addition to e-education, they call for further adoption of innovations towards being more of an immersive virtual learning environment. Learners in a virtual world lack autonomous pedagogical support such as autonomous feedback, expert guidance, and intelligent navigation support that should be available to them in the period they are performing learning activities. Furthermore, distant learners can feel isolated and bored if they can't find peers or tutors in the environment. Furthermore, in order to provide rich user interface with the 3D environment while providing rich functional support, innovations need to be considered to enrich the educational immersive experience. Adding intelligent, pedagogical, and autonomous support in the environment can be achieved through utilization of intelligent pedagogical agents (IPA's).

We have been investigating intelligent pedagogical agents for learning, finding several benefits to learning in 3D virtual learning environments, [4], [5], [6]. Intelligent pedagogical agents add values of pro-activeness, interaction, engagement of learners. Pedagogical agents can be central points of interaction and learning service providers for learners in those environments. When learners find autonomous learning support, they are encouraged to engage more in the learning process. Thus there have been studies on relevant aspects of IPA support to the learning process. For example, [7] reported that animated pedagogical agents increase learner performance through its *persona effect*.

Integrating an intelligent pedagogical agent in a virtual world requires different abilities including conversation, embodiment, animation, emotional abilities, and mediating interaction with environment. In general the IPA is to act as a central point of interaction between the environment and the learner to enrich the learning experience intelligently [8].

Design and implementation of intelligent pedagogical agents based learning in a virtual world has three thrusts of the implementation: the learner avatar aspects, the IPA aspects, and converting an object to become learning aware. IPA implementation aspects include several functions required of adopting lifelike character implementation, adding animation, and intelligence abilities. The learning object aspects as well as the learner avatar

require further considerations in order to add learning supported interactivity. For example, the learner requires further support and explanations about the learning object and the IPA requires the learning object to be controllable by it so as to provide step-wise guidance to the learner. In addition, the virtual world environment supporting the IPA operations mandates implementation efforts and experience targeting specific platforms and particular surrounding technologies. With those requirements, the realization of IPA in the virtual world faces the following challenges in the pragmatic dimension:

1. Experimentation learning of the virtual world technologies and selection of a practical virtual world platform. Several virtual world platforms exist such as the Open Simulator, [9] and Open Wonderland, [10].
2. Architectural, development, and implementation challenging aspects with the selected virtual world, [11].
3. Interfacing requirements from the virtual world to external environments such as an agent platform.
4. The implementation of an IPA imposes requirements of the IPA to be realized in the virtual environment including animation and multi-modal conversation abilities. This is in addition to the interaction with complex learning resources that have been designed for the purpose of deployment in the virtual world such as experiments found in [1] and [2]. In a virtual world, special interactivity of learning objects is required [12]. The IPA in an immersive learning environment requires understandable and controllable learning objects to serve its purpose in conducting tutorials.
5. The implementation of virtual worlds is at developmental stage especially for open source environments. For example, while the Non Personal Character (NPC) module in Open Wonderland is appealing for experimentation on the topic, it is categorized as “unstable”<sup>1</sup>.

In addition, realizing the IPA in the virtual world requires decision and reasoning abilities to provide expected intelligence. Intelligent agents add intelligence and reasoning support to the pedagogical agent taken from distributed AI research. Not only the agent approach provides the cognitive abilities, but also realizes further pedagogical functions. The pedagogical functions of intelligent agents with potential support in virtual learning environments are investigated in [4]. Moreover, in the intelligent agent model, several grouped agents suggest the ability to interact forming agent societies that exhibit social behavior to support collaborative learning functions [6][6].

Thus, design and implementation of intelligent pedagogical agents also entail investigating intelligent agents and their frameworks from AI research while their implementation in virtual worlds entail difficulties mentioned above. The intelligent agent part itself can be a basis for the design and development of IPAs providing interesting intelligent behavior such as autonomy, goal-directed behavior as well as re-activeness. Furthermore, research in intelligent agents strongly suggests the Belief-Desire-

Intention (BDI) model for action reasoning, [7][14][15] generally for cognitive layer function addition to pedagogical agents, [5]. However, the use of this special model raises several questions in comparison to design with traditional methods such as the object-oriented model. For example, how to set agent goals, how to set Beliefs, Desires, and Intentions? Which agent framework to choose, what design methodologies are suited? What could be learning scenarios with those agents, and how to design and implement BDI-based pedagogical agents for learning scenarios towards situating them in 3D virtual worlds?

Therefore, an agent-based simulation is required so as to investigate agent behavior not only to answer design questions, but also to isolate implementation challenges while discovering interactive and intelligent learning possibilities. And show how the pedagogical agent can support learning in a similar setting. Furthermore, interfacing the simulation with the prototype of IPA in the virtual world is required from the agent environment side in this context.

This paper is organized as follows. Section 2 discusses the BDI model of intelligent agents and its features contributing to pedagogical agent design. Section 3 discusses a learning scenario with pedagogical agent in a virtual world and its simulation in the agent environment. Section 4 discusses learning scenario implementation with two different agent platforms and illustrates an interface. Section 5 compares differences of the two platforms used and add extra aspects to consider for adopting a platform for pedagogical agent design. Section 5 is a conclusion.

## II. BDI AND GOAL DIRECTED BEHAVIOR FOR INTELLIGENT PEDAGOGICAL AGENTS

Designing intelligent pedagogical agents mandate the three properties; intelligence, pedagogical properties, and being an agent. The agent notion yields the meaning of an agent that is differentiated from a regular software objects by its proactive ability to act in an environment, [14]. The intelligence ability is pursuing AI different intelligent tactics supporting an objective and constructing intelligent behavior. Being pedagogical means to pursue effective pedagogical strategies and adopting its concepts. Hence, one way to look into this problem is applying AI intelligence of agents adding learning functions and strategies and finding a suitable agent framework for implementation and experimentation. From research, there have been several demonstrated benefits to incorporating pedagogical agents into virtual learning environments contributing to achieving pedagogical goals. The believability of the pedagogical agent leads to more engagement and interaction in the learning process. In the effort to more engagement of the learner, adding intelligent actions and services of the pedagogical agent can add to the desired believability and increased interaction. One approach to achieve it is through goal directed, autonomous, and pedagogical-aware behavior of the IPA by incorporating such methods from intelligent agents’ research. For implementation and experimentation, finding a practical agent platform for implementation and experimentation is required.

In agent research, intelligent agents provide three types of actions: reactive, proactive, and social behavior. On the other hand, the formation of a BDI system incorporates the design of plans, goals, events, and beliefs. Plans

<sup>1</sup> Unstable modules are: “either examples or experimental modules: there is no guarantee they will work at any point in time”, [13].

represent a manifest of what the agent should do and according to a situation that arises in the environment. When an event occurs, an agent plan can execute a course of action which represents a reactive behavior or contribute to achieving a goal. Agent reactive behavior in a learning scenario is useful to provide responses with or without reasoning to events in the environment or as a result of interacting with a learning object. For example, is to identify the learner's errors or to provide feedback.

Let's consider the proactive behavior of intelligent agents as goal directed behavior. One major need to intelligent pedagogical agents acting in a virtual learning environment is the ability to demonstrate goal directed behavior for several reasons:

- To pursue pedagogical goals in a non-deterministic environment. Our purpose is to utilize a 3D virtual world such as Second Life or OpenWonderland with adding intelligent pedagogical agents. Those environments are characterized by their scalability, availability of vast resources, and several users interacting through avatar representations making it non-deterministic in nature.
- To provide behavioral goals that mimic human behavior or provide advanced forms of cognitive abilities that lead to more believability and engagement in the learning environment
- A virtual embodied agent needs autonomy property which is achieved by dynamic goal-directed behavior. Goals are generated and achieved while learners are interacting in the environment. This property gives the autonomy property to the virtual teacher while providing learning services to the learners who are represented as avatars in a virtual world.

In agent based systems, a goal is a desire that the agent will pursue. It is rooted to human behavior as how intentions are formed based on work in [15]. Reasoning processes influence goals generation, how they are achieved, and can generate sub-goals in consequence. This reasoning is performed at run-time that gives its suitability to non-deterministic environments such as an educational virtual world.

It has been shown in research that the BDI model adds the cognitive layer in the design of the pedagogical agent [4][5]. Our particular interest is to answer the question of how the BDI model is realized upon implementing intelligent pedagogical agents with intelligent agent frameworks.

Furthermore, the reasoning method of BDI intelligent agents is appealing to be incorporated for pedagogical agents design. In the BDI based design, a hierarchical planning approach can be taken to divide the plan into smaller modules. Thus it is for reconstructing pedagogical goals into sub-goals with alternative smaller plans of achievement. The reasoning process will select the plans to execute only at run time with deliberation, [14]. That establishes a difference from traditional planning to agent-based planning that makes it suitable to add to the dynamic virtual learning environment. The selection though can be pedagogical-aware. Furthermore, dividing the plan into smaller plans can yield to generate different pedagogical permutations that can be suitable to different styles, environment situations, assessment and so on. This variation can follow the permutations of control model

suggested in [16]. Having such BDI controlled plan permutations adds pedagogical intelligence to the learning situation. This gives the pedagogical agent more decision alternatives to choose from so as to increase pedagogical functionalities such as adaptation.

Thus a BDI based implementation can also help in discovering new possibilities of adding intelligent pedagogical behavior to the IPA. The target is achieving the supporting functions in the virtual world. Simulation of potential BDI-based scenarios is required while being linked to its realization in the virtual world.

### III. SIMULATING A LEARNING SCENARIO WITH AN INTELLIGENT PEDAGOGICAL AGENT

A learning scenario in a virtual world is described in the following section followed by its corresponding simulation in the agent environment.

#### A. IPA-Based Learning Scenario in a Virtual World

Figure 1. Shows a learning scenario realized based on Open Wonderland virtual world environment [11] including an IPA, learner (represented as an avatar) and a learning object (device simulator). The learning object can be one of several experimentations available for online labs in a virtual world such as in [1] and [2]. In lack of a pedagogical agent, the learner has to interact with the learning object with no guidance or pedagogical support. Adding the pedagogical agent allows tutorials, mentoring, and assessment aided by different supporting services to learning in the context of the learning object. The IPA is equipped with modules to provide functions such as conversation and abilities to know about the learner. Proper interaction with the pedagogical agent leads to learning experience enrichment to the learner adding engagement, believability, and the opportunity not to feel isolated in the environment. Furthermore, the IPA can provide learning tracking services.

Other aspects of interaction among the three elements of IPA, learner avatar, and the experiment device are discovered in the agent-based simulation.

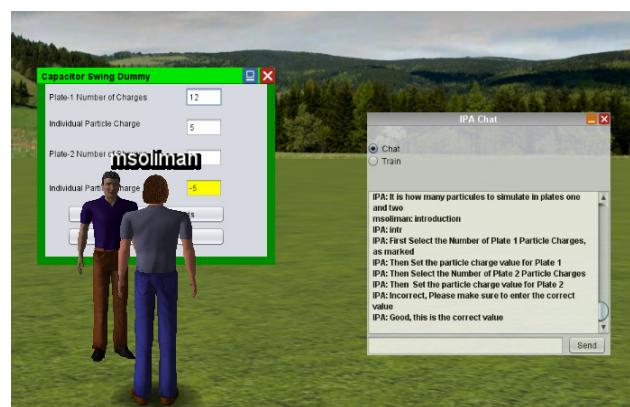


Figure 1. An IPA, a learner avatar, and a learning object (experiment simulation) in an interactive learning scenario in Open Wonderland. The IPA intervenes Learner-learning object interactions to provide learning support.

#### B. Agent-based Learning Scenario Simulation

There are three benefits to simulating learning with an IPA in the agent environment. 1) to stress intelligent agent interaction with the learner avatar given a learning object,

2) to isolate implementation challenges and efforts of the virtual world, and 3) to enable further agent-based extensions to learning. Prior research work in [4] and [6] refer to different learning possibilities with intelligent agents support. The simulated learning scenario involves three agents: an intelligent pedagogical agent, an agent to represent the learner avatar, and a device agent representing an experiment device simulation, see Figure 2.

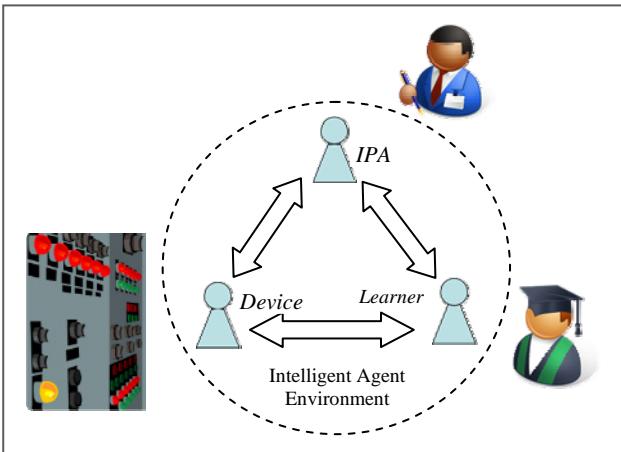


Figure 2. Three agents in the Agent Environment: Learner Agent, IPA, and Device agent. The IPA intervenes Learner-Device interactions.

In regular learning scenario settings, a learner interacts with a learning object conducting a simulation of a device operating the experiment through a control panel that will in turn provide parameters to a simulation to result in displayed output or 3D visualization in the virtual world. In this case, there is no guidance but a possible teacher avatar with regular virtual world settings. However, the regular teacher avatar lacks important function of availability 24/7, scalability for large number of potential users, and most importantly is not designed so as to understand or alter its operation. In simulating the virtual world scenario, the intelligent pedagogical agent is added in a way that intervene the interaction between the learner and the device to mediate the learning functions.

Upon the learner interaction with the learning object, the pedagogical agent observes learner contact with the device. At particular instance when the learner is responding incorrectly to a device observation, the IPA records it in the learner record and initiates a sub-goal so as the learner masters this particular situation. The purpose of the pedagogical agent is to add the following design objectives:

- Provide support to the learner through a tutorial.
- Monitor learner interaction with the learning object while assessing the learner and updating the learner abilities in regards to the activity.
- Provide immediate feedback. Providing such feedback has a special requirement to intervene according to environment or learner-learning object specific interactions. In the virtual world setting, the feedback is not only textual, or verbal, but can be emotional as well through IPA gestures.
- Intervene in situations when the learner is interacting incorrectly with the learning object so as to mitigate incorrect operation of the learning object. Early correction in a learning situation can be favored than

waiting till the end so that the IPA provides an immediate feedback.

- Control the learning object so as to generate specific behavior to check and assess the learner response.

This learning scenario is important compared to a one without a pedagogical agent, as shown, so as to provide a step-by-step guidance to the learner, improve engagement, add interactive assessment abilities, and provide facilities for integration with learning paths and other learning objects. This can be facilitated by a multi-agent implementation.

#### IV. INTELLIGENT AGENT-BASED IMPLEMENTATION

In order to implement the learning scenario with intelligent agents, a BDI-based agent system is needed. Several agent platforms such as JADE [17], Jadex [18], JACK [19], Jason [20], 2APL [21], and GOAL [22] are good candidates for implementation. Several of such environments are used to develop and integrate agents for virtual worlds, [5]. However, the selection and implementation are not trivial and hence experience and evaluation information of specific platforms and relevant issues are reported. One major obstacle in the selection is the lack of experience with those frameworks that have differences in implementation and the challenge in designing BDI-based agents that is non-deterministic by nature. An ease of use (pragmatic) factor such as tool support can give a good start to the novice developer who considers it as an important factor. While JACK [19] is a commercial platform, it provides graphical tools that can facilitate easy development and generation of intelligent agents. Then further criteria of evaluation can be investigated. [23] provides details of evaluation criteria and results concerning those agent platforms. While [23] resulted in good scores for JACK in the evaluation result, we are also interested in a non-commercial agent platform.

##### A. JACK based Implementation

JACK [19] is a java-based framework for building BDI based intelligent agents. JACK is characterized by a set of graphical development tools that help in visualizing the designs and interactions of the components and facilitate the design. JACK is composed mainly of agents, plans, events, and beliefs. JACK is based on the JACK Agent Language (JAL) being equipped with the graphical environment JACK Development Environment (JDE). JAL extends Java syntax to allow BDI concepts such as reasoning methods. For example is to define plans processing rules of generated events.

Implementing the learning scenario with JACK involves creating three agents; pedagogical agent, learner agent, and a learning object. The role of the pedagogical agent is to direct and monitor learner to learning object interaction. Since we are not implementing it directly to a virtual world, the purpose of the learner agent is to simulate the learner avatar. The purpose of the learning object agent is to simulate a visual learning object that can be a physics experiment running in a virtual world as discussed. The simulation allows running the device with simple controls that results in different observations. The experiment runs as the pedagogical agent asks the learner to run the experiment or provide a step-wise instruction. At certain steps, the pedagogical agent can intervene so that the device generates an alternative behavior (observa-

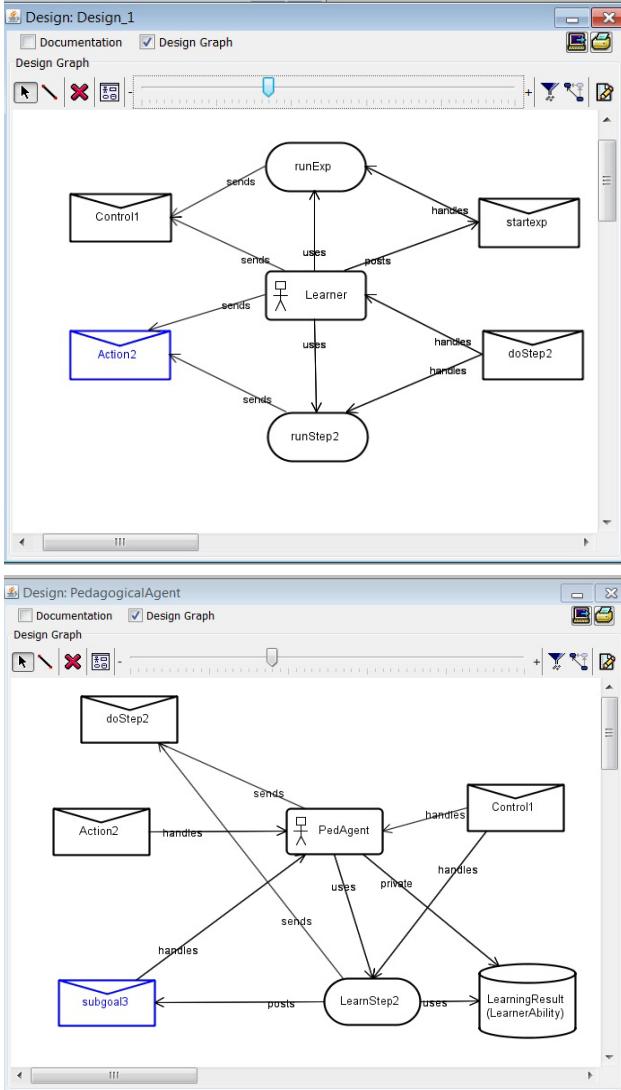


Figure 3. Example design diagrams of two agents in a learning scenario. The envelope shape represents an event while the oval represents plans.

tion) then the learner should do different interaction to the device. Upon incorrect action, the pedagogical agent will update the learning result belief base. Figure 3. illustrates JACK design diagrams of the learner agent and the pedagogical agent. Events generated are sent, posted, or handled by different plans.

#### B. Jadex based Implementation

Jadex [18], [24] has similar BDI concepts to JACK in agent plans, goals, events, and beliefs. Jadex is built on top of JADE, the FIPA compliant agent platform. Communication in Jadex follows the Agent Communication Language (ACL) FIPA standard as well, [25][18]. A major characteristic of Jadex is the XML based Agent Definition File (ADF) that describes the agent, goals, and beliefs, see Figure 4. A plan in Jadex is a Java-based program. There are two types of plans in Jadex; a service plan that handles multiple events, and a passive plan for each event. Events can be message events, goal events, or belief updates. Figure 5. Shows plan definition and two event types in a Jadex implementation of our learning scenario. In Jadex, there are four types of goals: 1) achieve, 2) query, 3) maintain, and 4) perform goals. An

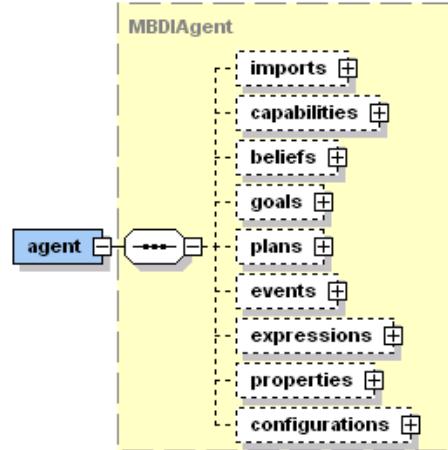


Figure 4. Jadex Agent Definition File Specification (Source: [18] )

```
<!-- A Learning plan - step 2 -->
<plan name="runStep2">
  <body class="Step2actionPlan"/>
  <trigger>
    <messageevent ref="do_step2"/>
  </trigger>
</plan>
</plans>

<events>
  <!-- 1. action request do step 2 of the experiment initiating this
  plan
      with perfor. request. 2. learner performs action2, internal ev. --
  >
  <messageevent name="do_step2" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
      <value>SFipa.REQUEST</value>
    </parameter>
  </messageevent>

  <internalevent name="action2_update">
    <parameter name="content" class="String"/>
  </internalevent>

```

Figure 5. Code snippet from Jadex agent ADF showing plan definition and two types of events; plan triggering event and an internal event.

agent, defined in an ADF, along with the java plan run through the Jadex Control Center (JCC).

Similar to JACK, three agents to represent the simulation are created: pedagogical agent, learner, and device simulator as separate ADFs with the corresponding plans. One major difference from JACK is in the inter-agent communication between different agents. In JACK, an event can be processed from another agent plan than the one that has generated it as long as it is declared to be handled. In Jadex, events are handled (listened) by different plans but for the same agent as long as inter-agent communication is handled through ACL message passing. This issue needs to be taken care off in the design of the Jadex agents.

#### C. Interfacing to the Agent Environment

Furthermore, interfacing to Jadex agents can be achieved through HTTP request to a server that is listening for a port. This motivates a design of an interface agent that listens with its service plan to HTTP requests and forwards the request to the corresponding agent. Since the forwarding mechanism is through different agents, it is by ACL messages. Furthermore, a sub-goal is triggered

upon receiving an event to further process it in a goal-based manner. For example, to interpret events based on agent beliefs, see Figure 6. The interpretation can provide the relation of the event of interest to which agent or it can generate an event in the simulation agent environment.

## V. EVALUATION AND FURTHER CONSIDERATIONS WITH THE AGENT ORIENTED IMPLEMENTATION

A simulation of a learning scenario is provided that include an interface agent. Further aspects of concern include how to integrate the agent platform with a virtual learning environment to allow agents to send actions to the environment and perceive events from it. This integration is not trivial given a large number of events the environment generates while not of interest to the agent platform [26] and the differences of the VW design and the agent design [27]. In agent systems, a percept defines methods to send and to receive events of interest and manage them between the two environments which are not trivial considering specific VW implementations. A major aim of the Environment Interface Standard (EIS) [28] is to standardize the interaction between the agent platform and the environment which is a virtual world in our case. An interesting feature is its ability to link to connect an agent platform to any environment that implements the EIS. Jadex is supported by EIS as reported in [28]. Jadex also provides environment support through EnvSupport for developing virtual environments, see Figure 7.

We have modeled the learning object in the above scenario as an agent so as to contribute to reasoning. Adding objects and environment behavior and interaction into the reasoning can yield better goal attainment and further intelligence. But it adds requirements to the selection of the agent platform and the ability of the environment to provide agent-aware objects. How the environment can provide more flexibility for assigning and achieving pedagogical goals? And how objects are designed to contribute to pedagogical goal attainment? The Common Artifact infrastructure for Agents Open environments (CARTAgO) [29] adds this dimension by suggesting an agent-aware approach to engineer environment objects (artifacts) based on the activity learning theory. It suggests the Agents and Artifacts (A&A) paradigm considering the cognitive aspects of learning objects to engineer artifact-based environments. C4Jadex has been developed as a bridge for Jadex platform to allow such integration to CARTAgO, [30], [31].

On the other hand, JACK provides interesting features for group reasoning and team oriented modeling by using JACK Teams. Furthermore, CoJack is a BDI cognitive architecture extends on JACK for modeling human behavior thus allowing for humanoids/virtual actors development.

## VI. CONCLUSION

This paper discussed simulation of learning scenarios with agent-based intelligent pedagogical agents in relation to virtual world realization. The simulation provides a proof-of-concept of the agent-oriented behavior in learning scenario setting providing details of IPA supported interaction. A BDI based implementation is provided based on three agents: IPA, learner agent, and device agent. In the scenario of the implementation, the IPA

```
<beliefs>
  <!-- This beliefset contains the dictionary as facts. -->
  <beliefset name="events" class="Tuple">
    <fact>new Tuple("event1", "Go")</fact>
    <fact>new Tuple("event2", "Dont Go")</fact>
  </beliefset>
</beliefs>

<goals>
  <!-- goal containing the socket for client communication. -->
  <achievegoal name="checkEvent">
    <parameter name="Client" class="Socket"/>
  </achievegoal>
</goals>

<plans>
  <!-- Passive plan for processing an event from event.
      Selectively Reacts on changes made based on the belief set clients. -->
  <plan name="eventHandler" class="EventPlan">
    <parameter name="client" class="Socket"/>
    <goalmapping ref="checkEvent.client"/>
    </parameter>
    <body class="TranslationPlan"/>
    <trigger>
      <goal ref="checkEvent"/>
    </trigger>
  </plan>
  <!-- Initial plan for starting a server thread waiting for client connections.
      Adds new sockets with the new connection to the client belief set. -->
  <plan name="server" class="ServerPlan">
    <parameter name="port" class="int">
      <value>9099</value>
    </parameter>
    <body class="ServerPlan"/>
  </plan>
</plans>
```

Figure 6. Snippet from the Agent Definition File (ADF) for a Jadex Interface Agent

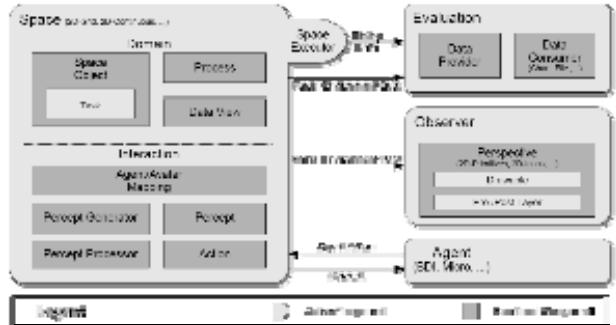


Figure 7. Main conceptual building blocks of EnvSupport in Jadex (Source: [18])

provided different supporting functions including tutoring, monitoring, feedback, correcting learner behavior, and assessment. The IPA agent alters the behavior of the learning object so as to assess and train the learner on dealing with different outcomes of an experiment.

The agent-based implementation yields identifying plans, goals, and beliefs following the BDI model. In this regard, the paper illustrated JACK and Jadex based implementations of the simulation scenario. Comparison of both platforms shows similarities in the agent functions in goals, plans, and beliefs. Differences of both platforms reported are differences in the representation of the agent through ADF, open source ability, standardization, availability of graphical design tool, and extensions. Further potential supporting functions include teams support, plan representation, supported goal types, environment support, and understanding ability and its impact on manipulating learning objects.

In order to implement the learning scenario in a virtual world, interfacing to the agent environment is required to perceive environment events and provide results back as forms of actions to the environment. The paper discussed integration aspects of the both environments reporting efforts that attempt to standardize integration to multiple agent platforms. An interface agent is shown in the Jadex agent environment based on HTTP requests and replies.

It is found that this simulation approach not only isolates implementation efforts and challenges in the virtual world but also stresses intelligent-agent based interaction in this setting and enables opportunities for further discovery.

## PAPER

### SIMULATING INTERACTIVE LEARNING SCENARIOS WITH INTELLIGENT PEDAGOGICAL AGENTS IN A VIRTUAL WORLD THROUGH BDI-BASED AGENTS

#### REFERENCES

- [1] B. Scheucher, P. Bailey, C. Guetl, and V. J. Harward, "Collaborative Virtual 3D Environment for Internet-accessible Physics Experiments," International Journal of Online Engineering (iJOE), Vol. 5, pp. 65–71, 2009.
- [2] J. Pirker, S. Berger, C. Guetl, J. Belcher, and P. Bailey, "Understanding Physical Concepts using an Immersive Virtual Learning," In Proceedings of 2nd European Immersive Education Summit, 26-27 Nov. 2012, Paris.
- [3] Immersive Education, <http://immersiveeducation.org/>, Online, Accessed Dec. 15, 2012.
- [4] M. Soliman and C. Guetl, "A survey of pedagogical functions of intelligent agents in virtual learning environments ", Journal of Internet Technology (JIT), Special Issue on Agent Technology in Cyberspace, Vol 12, No 6, 2011.
- [5] M. Soliman and C. Guetl, "Evaluation of intelligent agent frameworks for human learning in virtual worlds," International Journal of Engineering Pedagogy (ijEP), Vol 1, No 3, 2011.
- [6] M. Soliman and C. Guetl, "Review and perspectives on intelligent multi-agent systems' support for group learning," World Conference on Educational Multimedia, Hypermedia & Telecommunications, ED-MEDIA 2010, June 2010, Toronto, Canada.
- [7] J. Lester, S. Converse, S. Kahler, T. Barlow, B. Stone, and R. Bhogal, "The Persona Effect: Affective Impact of Animated Pedagogical Agents," In Proceedings of CHI '97 (pp. 359-366), Atlanta, March 1997, Miami, Florida, USA. <http://dx.doi.org/10.1145/258549.258797>
- [8] M. Soliman and C. Guetl, "Intelligent Pedagogical Agents in Immersive Virtual Learning Environments: A review," Proceedings of 33rd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, May 24-28 2010, Opatija, Croatia.
- [9] The OpenSimulator, <http://opensimulator.org>, Online, Accessed: Dec. 15, 2012.
- [10] OpenWonderland, <http://openwonderland.org/>, Online, Accessed Dec. 15, 2012.
- [11] J. Kaplan and N. Yankelovich, "Open Wonderland: An Extensible Virtual World Architecture," IEEE Internet Computing, 2011, 38-45. <http://dx.doi.org/10.1109/MIC.2011.76>
- [12] P. Jorissen and W. Lamotte "A Framework Supporting General Object Interactions for Dynamic Virtual Worlds," In Proceedings of the 4th International Symposium, SG 2004, Banff, Canada, May 23-25, 2004.
- [13] Open Wonderland Project Modules, <http://code.google.com/p/openwonderland/wiki/DownloadBuildModules05>, Accessed Dec. 15, 2012.
- [14] M. Wooldridge. An Introduction to MultiAgent Systems. John Wiley & Sons, 2002.
- [15] M. Bratman. Intention, Plans, and Practical Reason. Harvard University Press. Cambridge, MA, USA. 1987.
- [16] A. Baylor. "Permutations of control: Cognitive considerations for agent-based learning environments," Journal of Interactive Learning Research. 12(4), 403-425, 2001.
- [17] JADE, Java Agent DEvelopment Framework, <http://jade.tilab.com/>, Online, Accessed Dec. 15, 2011.
- [18] Jadex, <http://jadex-agents.informatik.uni-hamburg.de> , Online, Accessed Aug 31, 2012.
- [19] JACK, <http://aosgrp.com/products/jack/index.html>, Online, Accessed Aug. 31, 2012.
- [20] Jason, <http://jason.sourceforge.net/wp/> , Online, Accessed Dec. 15, 2012.
- [21] M. Dastani, "2APL: a practical agent programming language," International Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), 16(3):214-248, 2008. <http://dx.doi.org/10.1007/s10458-008-9036-y>
- [22] GOAL Agent Programming Language, <https://mmi.tudelft.nl/trac/goal/>, Online, Accessed Dec. 15, 2012.
- [23] L. Braubach, A. Pokahr, and W. Lamersdorf. "A universal criteria catalog for evaluation of heterogeneous agent development artifacts. In: Proc. of AT2AI-6, IFAAMAS, 2008.
- [24] L. Braubach, A. Pokahr, and W. Lamersdorf, "Jadex: A BDI agent system combining middleware and reasoning," in Software Agent-Based Applications, Platforms and Development Kits. Whitestein Series in Software Agent Technologies. 2005, pp 143-168. [http://dx.doi.org/10.1007/3-7643-7348-2\\_7](http://dx.doi.org/10.1007/3-7643-7348-2_7)
- [25] Foundation for Intelligent Physical Agents, <http://www.fipa.org/>, Online, Accessed Dec. 15, 2012.
- [26] J. Oijen and F. Dignum. "Scalable perception for BDI-agents embodied in virtual environments," WI-IAT, vol. 2, pp.46-53, 2011. IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology. 2011.
- [27] J. Oijen, L. Vanhée, and F. Dignum. "CIGA: A middleware for intelligent agents in virtual environments," In Proceedings of the 3rd International Workshop on Agents for Education, Games and Simulations, AAMAS'11, Taipei, Taiwan.
- [28] T. Behrens, K. Hindriks, and J. Dix, "Towards an environment interface standard for agent platforms," Annals of Mathematics and Artificial Intelligence, 61, 4 (April 2011), 261-295. <http://dx.doi.org/10.1007/s10472-010-9215-9>
- [29] A. Ricci, M. Viroli, and A. Omicini. "CARTAgO: a framework for prototyping artifact-based environments in MAS, " In: Weyns, D., Parunak, H.V.D., Michel, F. (eds.) Environments for MultiAgent Systems III, pp. 67–86. Springer (2007). [http://dx.doi.org/10.1007/978-3-540-71103-2\\_4](http://dx.doi.org/10.1007/978-3-540-71103-2_4)
- [30] M. Piunti, A. Ricci, L. Braubach, and A. Pokahr, "Goal-Directed interactions in artifact-based MAS: Jadex agents playing in CARTAGO environments," In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2008), Sydney, NSW, Australia., 2008. <http://dx.doi.org/10.1109/WIAT.2008.349>
- [31] C4Jadex. <http://apice.unibo.it/xwiki/bin/view/CARTAGO/C4Jadex>, Online, Accessed Dec. 15, 2012.

#### AUTHORS

**M. Soliman** is with the Institute for Information Systems and Computer Media, Graz University of Technology, Graz, Austria (e-mail: Muhammed.soliman@gmail.com).

**C. Guetl** is with the Institute for Information Systems and Computer Media, Graz University of Technology, Graz, Austria. He is also adjunct research professor with the School of Information Systems, Curtin University, Perth, Western Australia (e-mail: Christian.Guetl@iicm.TUGraz.at).

This article is an extended and modified version of a paper presented at the International Conference on Interactive Collaborative Learning (ICL2012), held 26 - 28 September 2012, in Villach, Austria. Received 17 December 2012. Published as resubmitted by the authors 18 March 2013.