Proposal of an Architecture to build Intelligent Learning Objects based on BDI agents

Natanael R. Bavaresco¹, Ricardo A. Silveira²

¹ Federal University of Santa Catarina, Brazil, natanael@inf.ufsc.br
² Federal University of Santa Catarina, Brazil, silveira@inf.ufsc.br

Abstract: The learning objects are a great initiative to increase the reusability of educational material. They are able to provide important information to the teacher about the students. Despite this, the e-learning systems are not prepared to attend the individual learner’s characteristics. There is a need that these systems become more intelligent and more adaptative. This work presents a system based on BDI agents to support the e-learning and provide learning experiences. This paper proposes the design of a multi-agent system based on Belief-Desire Intention (BDI) paradigm to create Intelligent Learning Objects (ILOs). The system is supported by techniques of Software Engineering oriented to the agents for the development of agent society.

Keywords: Intelligent Agents, Learning Objects, Agent Oriented Software Engineering.

1. Introduction

This paper describes an application of a Learning System based on BDI Multiagent Systems [3] and Learning Objects to promote learning experiences. According to [14] and [8]; a Learning Object is an entity of instructional content that can be applied several times in different courses or contexts. Developing learning materials for e-learning can be expensive, but as the content of related courses taught at different universities and organizations often tend to be similar, the cost can be shared among its potential users [6]. The use of reusable learning objects to create learning environments improves the quickness, the flexibility and the economy. In order to be reused, a learning object must be modular, discoverable and interoperable [7]. To achieve these features and improve the efficiency, efficacy and reusability of learning objects, many people have dedicated a great effort to do definition of standardization. Organizations such as IMS Global Learning Consortium [9], IEEE [8] and [4], have contributed significantly by defining indexing standards called metadata.
Metadata structures contain the information to explain what the learning object is about, how to search, access, and identify it and how to retrieve educational content according to a specific demand.

However, there are limitations of current learning objects. The task of finding the right object may be quite time consuming because the course designer must carefully examine each learning object. In addition, current learning object metadata standards are not very useful in supporting pedagogical decisions [9].

We proposed a new version of the Intelligent Learning Objects (ILO) approach, developed in [15], based on BDI [3] agents’ architectures. The BDI agent-based architectures define beliefs, desires, and intentions as mental attitudes represented as possible world states. The use of this approach can attaches new features and behaviours to the learning objects as presented above for agent-based learning environments.

2. Learning Objects

The main focus about Learning Objects has been on the definition of standardization by defining indexing standards called Metadata. It explains basically what the learning object is about how to search it and how to retrieve educational content according to a specific demand.

2.1 SCORM Standard

The Learning Objects used are built according with the SCORM specification. This standard was selected because provide important guidelines.

- Reusability: it can be easy to update and works with any authoring tool. Also, it must be applied in many contexts;
- Accessibility: it refers the capability to have been found and became available to learners and developers in anywhere;
- Interoperability: it must be operable in different kind of hardware (operational systems and web browsers, for example);
- Durability: it will not be necessary change the learning object in case of new versions of software.

This standard describes a Content Aggregation Model (CAM), a Run Time Environment (RTE) and a Sequencing and Navigation Model (SN).

The content aggregation model aims to give a common concept to build learning content. Also, it defines that learning content must be identified and described, to be applied in a course, or transferred into LMS and repositories. The model includes specification to content and metadata. Some components are defined, such as: assets, shareable content object, activities, content organization and content aggregation.
The Run-Time environment is responsible for communication between learning object and LMS. Figure 1 shows a client-server architecture of the environment using a SCORM API Adapter [1]. API provides a standardized way for Shareable Content Objects (SCOs) to communicate with LMS. An active SCO can access a set of functions already defined. Hence, the LMS must have this adapter to run objects and access additional information, for example, student data model.

The Sequencing and Navigation describes a set of elements that may be used by content builders to define their intentions of behavior of sequence and delivery of activities. With these features, it is possible to retrieve learner data while the learning experience is happening.

3. BDI Agents

The BDI model represents a cognitive architecture based on mental states initially proposed by [5]. It consists of the concepts of belief, desire, and intention as mental states that generate human action. Beliefs capture informational states, desires motivational states, and intentions deliberative states of agents. These mental states can lead the course of agent actions. This process is called deliberation. The deliberation process consists in creating new intentions of the agent based on its beliefs, desires, and intentions. The process has two steps:

- Option Generation: Consists of the choice of a set of options (desires) observing the actual beliefs and intentions.
- Filtering: It’s the choice of the best option generated on the previous step, which will create the new intention.

3. Intelligent Learning Objects

An Intelligent Learning Object (ILO) is a kind of agent able to promote learning experiences to students in the same way as learning objects can do. For this reason, an ILO can also be seen as a learning object built through the agent paradigm. The technological basis of this approach is composed by a combination between technologies developed for Learning Objects and Multi-agent Systems. According to [16], an Intelligent Learning Object (ILO) can use the Agent Communication Language (ACL) for communication among other ILOs. This capability provides a powerful communication improved by the Content Language (CL) and Ontologies. By using the ACL, it is possible to communicate expressing facts, rules, or mental states. Another useful capability of the agents is the learning ability. An Intelligent Learning Object, with these features, may acquire new knowledge in its existence interacting with learners and other ILOs like: acquire new learning
materials and adapt their beliefs based on the individual learning styles and preferences of the student. Mental states based agents can perceive changes in the environment and can update beliefs in your belief base that can lead to different actions during their existence. This capability allows more dynamic interactions between ILOs and learners differing of the static interaction between common learning objects.

4. Modeling the Agents

The ILO agent society was built using the Prometheus [13] Agent Oriented Software Engineering. This methodology provides detailed support to development, specification and test. Prometheus is composed of three phases:

• The first concerns to system specification: The system environment, functionalities and goals.
• The second (architectural project) use the output of previous phase to determine which agents will exist in the system and how they will interact
• The last phase (Detailed Project) is responsible for defining capabilities of the agents, internal events, plans and a detailed data structure for each type of agent identified in the previous phase.

The Goal Diagram shows the goals of the Multi-agent system and the sub-goals that must be reached by the system.

Figure 1. System Goals Diagram
The central diagram is the System Overview Diagram, where it is possible to observe all agents and their roles, perceptions, messages and actions. This diagram provides the main functioning of the system.

5. Proposed Architecture

The architecture proposed in this work consists of a web server, a repository of learning objects, and an agent server. This way, the learner can access the system and request for a learning experience. The web server will host the web pages and servlets of the application. All the reasoning will be generated by the interaction among agents and through deliberation. The repository server will host the learning objects, and the agents may be able to search and retrieve the better object to a specific learner. The figure below shows an overview of the system.

![Figure 2. Multi-Agent System Overview](image-url)
6. Development

The SMA development was performed using Jadex framework 0.96. This framework provides basic agent services such as a communication infrastructure and management facilities. Through this framework it is possible creating BDI agents defining their beliefs, goals and plan heads in a XML file called Agent Definition File (ADF) and allows integration with Jade Framework [11]. The recipe to plan execution must be written in a Java file.

The integration with web systems and agents was performed by an add-on called Webbridge. This add-on allows the communication between agents and java servlets by a servlets called DelegateServlet that receive the learner request and deliver to agents through ACL messages. After this, a CoordinatorAgent, provided by the Webbridge, receives the request and forwards to the agent that will to be able to perform the necessary actions such as send messages and create new agents.

Figure 3. Overview of the Architecture

Figure 4. Jadex Webbridge
An architecture to create Intelligent Learning Objects based on BDI agents

The system has a servlet responsible for validating the user and then starts the learning experience. By logging in the system, the user have his information consulted by LMSAgent (an abstraction of Learning Management System) in the database that updates its beliefs, generate goals and start the plans executions like ask the ILORAgent (abstraction of Learning Object Repository) for learning objects. After this with the selected object, an ILOAgent (abstraction of a Learning Object) will be created who has the responsibility to display the object to the student, monitor their interaction and retrieve data through Scorm API. During the user interaction with the object the ILOAgent might understand that it is a necessary complement the learning or even a revision in the preliminary stages of the course that were previously completed. It is triggered an event that create exchange of messages between ILORAgent and LMSagent. The LMSAgent then updates the information from the apprentice and starts the cycle again requesting a new ILO.

![Figure 5. System Functioning](image)

7. Conclusions

This paper shows a system based on previously works like [10] and [15] proposing a new approach to the learning with ILOs using BDI agents. The BDI agents reasoning through mental states and might update their beliefs in execution time. This can result in a more interactive system, adapting the courses or guide the learner to a review of previous concepts.

The use of an agent-oriented software engineering methodology helps to establish the goals, roles, capabilities and interactions with the agents. The Prometheus Methodology is easy to understand and provides complete support to create BDI agents’ systems.
The Jadex framework provides a complete support to agent creation and monitoring. The agent definition file is easy to understand and enables the creation of agents with complex behaviours. The possibility of communicate with Java servlets enables the creation of friendly interfaces to user, abstracting the deliberation process behind of the agents.

For future work, the application will be continued, as additional tests with ILOs will be performed and the compliance with the SCORM standard.

References

