Improving Interoperability Among Learning Objects Using FIPA Agent Communication Framework

Ricardo Azambuja Silveira[1], Eduardo Rodrigues Gomes[2], Rosa Vicari[2]

1 Universidade Federal de Santa Catarina, Campus Universitário, s/nº -
Caixa Postal 476 Florian'opolis – SC, Brazil, silveira@inf.ufsc.br

2 Universidade Federal do Rio Grande do Sul, Av. Bento Gonçalves,
9500 Bloco IV Porto Alegre - RS –Brazil, {ergomes; rosa}@inf.ufrgs.br.

Abstract. The reusability of learning material is based on three main features: modularity, discoverability and interoperability. Several researchers on Intelligent Learning Environments have proposed the use of architectures based on agent societies. Learning systems based on Multi-Agent architectures support the development of more interactive and adaptable systems and the Learning Objects approach gives reusability. We proposed an approach where learning objects are built based on agent architectures. This paper discusses how the Intelligent Learning Objects approach can be used to improve the interoperability between learning objects and pedagogical agents.

1 Introduction

This paper addresses the improvement of interoperability among Learning Objects in agent-based Learning Environments by integrating Learning Objects technology and the Multi-Agent Systems approach. A Learning Object, according to Downes (2001, 2002), Mohan & Brooks (2003), and Sosteric & Hesemeier (2002), is a piece of learning content that can be used several times in different courses or in different situations. According to Downes (2001), the expense of developing learning materials for e-learning can be large, but as the content of related courses tend to be similar, the cost of developing the learning material can be shared. The learning object approach promises to reduce significantly the time and the cost required to develop e-learning courses. The design of learning environments using reusable learning objects improves quickness, flexibility and economy.

A learning object must be modular, discoverable and interoperable, in order to be reused (Friesen, 2001). To achieve these features and improve the efficiency, efficacy and reusability of learning objects, many people have dedicated a great effort. The main focus has been on the definition of standardization. Organizations

such as IMS Global Learning Consortium (2004), IEEE (2004), ARIADNE (2004), and CanCore (2004), have contributed significantly by defining indexing standards called metadata (data about data). Metadata contain 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.

Mohan & Brooks (2003) point out the limitations of current learning objects. According to them, an instructional designer must carefully examine each learning object to find the right object. This job may be quite time consuming and the learning object metadata are not very useful to support this task.

On the other hand, the state of the art in Intelligent Tutoring Systems (ITS) and Intelligent Learning Environments (ILE) points to the use of Agent Society-Based Architectures. Multiagent Systems (MAS) have proved to be appropriate for designing tutoring systems because learning is dealt with as a cooperative task (Johnson & Shaw, 1997). Using the MAS approach in designing ITS and ILE can result in faster, more versatile and low cost systems. The agents composing such systems use to be called Pedagogical Agents (Johnson & Shaw, 1997).

In Silveira et al (2005), we proposed the development of learning objects based on agent architectures: the Intelligent Learning Objects (ILO) approach. The use of agent-based architectures gives to the learning objects, the same features as presented above for agent-based learning environments.

2 Intelligent learning objects

An Intelligent Learning Objects (ILO) is a kind of agent able to promote learning experiences to students as 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 Multiagent Systems.

There are many benefits of integrating learning objects and agents: An Intelligent Agent is a piece of software that works in a continuous and autonomous way in a particular environment, generally inhabited by other agents, and able to interfere in that environment, in a flexible and intelligent way, not requiring human intervention or guidance (Bradshaw, 1997). An agent is able to communicate with others by message exchange using a high-level communication language called Agent Communication Language (ACL), which is based on Logic concepts.

We have started from the learning object model called SCORM (ADL, 2004). SCORM performs the communication by calling methods (functions) and passing parameters, according to the Object Oriented Programming paradigm. An ILO uses ACL for communication among learning objects, hence the learning environments can perform a more powerful communication. ACL gives a more powerful semantic in communication using a formal protocol and a formal Content Language (CL) based on some logic formalism to express the messages content. By using ACL, it is possible to communicate not only variable values, but also facts, rules, mental states

and more. The result is that communication by ACL and CL is potentially much better than the calling methods like in the object oriented approach.

The potential learning ability of intelligent agents. gives to the ILOs the ability to acquire new knowledge and perform different behaviors during its existence, as according to its own experience. Thus, by interaction with students and other ILOs, an ILO is able to evolve. It is not a static piece like current learning objects.

Agents can have coordination and cooperation mechanisms that help the agents` society to achieve its goals. Such agent features can be very useful due to the a self-organizing ILO society where it can promote richer learning experiences. The coordination and cooperation mechanisms enable complex behaviors and interactions among ILOs and, as a consequence, more powerful learning experiences.

Other agent features that promote interaction among learning objects are autonomy, pro-activity, sociability and benevolence. The autonomy of ILO gives it the capability to act based on its own behavior and knowledge with no external intervention. The pro-activity feature assures that the ILO must act in order to satisfy its own objectives. The sociability and benevolence features address the capability of social and cooperative behavior.

As a learning object, an Intelligent Learning Objects must be reusable. The *reusability* is given as a result of three features: interoperability, discoverability and modularity (Friesen, 2001). In Learning Objects approach, the use of metadata to describe the pedagogical content of the learning object gives discoverability. To enable this feature in ILO, we adopted the IEEE *1484.12.1 Standard for Learning Object Metadata* (LTSC, 2004). The *modularity* of learning objects can be reached with a good pedagogical project. So, the design of the pedagogical task of an ILO must be made according to the expertise of some object matter specialists and pedagogical experts. Some interoperability can be reached by the use of well-known standards. For this reason, we adopted two learning object standards: a) the IEEE *1484.12.1 Standard for Learning Object Metadata* (LTSC, 2004); and, b) *IEEE 1484.11.1 Standard for Learning Technology – Data Model for Content Object Communication*. The 1484.11.1 standard is defined for communication of learning objects with *Learning Management Systems* (LMS). We use this standard in interactions among ILOs. In order to assure interoperability among agents we have adopted the FIPA (2002) reference model and FIPA-ACL; as the language to be used for communication. We used these technologies to define a communication framework for ILOs. The ILOs must use this framework in order to communicate with each other.

The *discoverability* is the ability to be discovered in terms of its tasks and the services it provides. In addition to some services provided by the FIPA architecture, our communication framework contains a set of dialogues that ILOs should use. Besides those technological issues, one important requirement for an ILO is that it must have an educational purpose. Thus, an ILO must be created and applied in order to carry out some specific tasks to create significant learning experiences by interacting with the student. For this reasons, the project of an ILO needs a contents expert, an educational expert and a computing expert.

3 The ILO Multi-Agent Architecture

The agent society presented by Silveira et al (2005) encompasses three types of agents: *LMS agents* and *Intelligent Learning Objects agent* and *ILOR Agent*.

   Intelligent Learning Objects are agents responsible for playing the role of learning objects. Its responsibility is to generate learning experiences to the students in the same sense of learning objects. *LMS Agents* are abstractions of Learning Management Systems. It is responsible for dealing with the administrative and pedagogical tasks involving a learning environment as a whole. It provides a way for students to access ILOs, and get information concerning the studentsto the LMS, *ILOR Agents* are abstractions of Learning Objects Repositories systems. It is responsible to store data about he ILOs to permit a user or an agent to find them. Figure 1 illustrates the proposed agent society. First, students interact with the LMS Agent in order to have learning experiences.
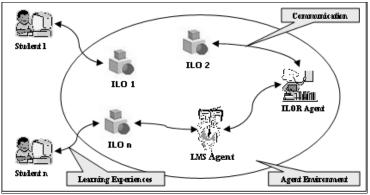

Fig. 1. Proposed agent society

   The LMS Agent searches (with the aid of the ILOR Agent) the appropriate ILO and invoke it. The ILO is then responsible for generating learning experiences to the students. In this task it can communicate with the LMS agent and with other ILO agents in order to promote richer learning experiences. All the communication is performed by messages exchange in FIPA-ACL. The agent environment that these agents inhabit is a FIPA complaint environment. It provides all the necessary mechanisms for message interchanging among the agents.

3.1   Agent Communication structure

One of the main concerns of this research is the modeling of the communication processes among the agents. Through a well-defined communication framework it is possible to improve interoperability because it enables different types of agents to share information.

We defined a communication framework based on FIPA reference model (FIPA, 2002). FIPA uses the idea of communication as the exchange of declarative statements. In this kind of communication, agents receive, reply and send requests for services and information transported by messages. There are five main concepts: Agent Communication Languages (ACL), Content Languages (CL), Agent Interaction Protocols (AIP), Conversations and ontologies. An ACL is responsible for defining how the contents of a message have to be interpreted. A CL is a declarative knowledge representation language to encode the message's content. An AIP is a typical communication pattern with an associated semantic to be used by the agents. A Conversation occurs when an agent instantiates an AIP in order to communicate with other agents. Finally, an Ontology defines the terminology used to denote domain-specific concepts in the message's content. In this section we show the communication structure that must be used to build Intelligent Learning Objects.

3.2 Interaction protocol

The FIPA reference model presents a set of interaction protocols. In this work we use the FIPA-Request protocol (FIPA, 2002), which is the FIPA protocol that should be used for agents who want to do requests for services provided by other agents.

The FIPA-Request protocol begins with a *request* message denoting that the sender agent asks the receiver agent to perform the task defined in the content of the message. The content of the message is an *action* describing the task that the receiver agent is supposed to do. An action is an abstraction of a real concept of an action that an agent can execute. Its semantic is defined in an ontology. For example, the action *send-metadata,* defined in our ontology, can be used by an agent to ask the metadata about an ILO. If the receiver agent agrees to perform the requested task, the final message will be an *inform* containing a *predicate*. A predicate says something about the state of the world and can be true or false. In this work we used three predicates defined by FIPA:

a) *result*, which means that the result of the action pointed in the first argument is contained within the second argument;

b) *done*, which means that the action in its first argument is done and no result was returned; and

c) *set*, which introduces a set of concepts or predicates.

3.3 The ontology for intelligent learning objects

For the FIPA reference model, ontology is a set of domain-specific concepts within the messages contents. A typical FIPA ontology is defined by using predicates, actions and concepts (FIPA, 2002). Although simple, this definition is very pragmatic and satisfies all the requirements of the agent communication processes defined in this work.

This section shows the actions and concepts which compose the ontology used in this framework. As we used FIPA-SL0 (a sub-set of FIPA-SL) to encode the messages' content, this section also shows how these items shall be represented using FIPA-SL0. We are using **bold** to describe terminal symbols and *italic* to

describe non-terminal symbols. Symbols defined in the ontology itself are enclosed by "<" and ">"

In addition to the following ontology, the agents must know the ontology defined in the document SC0000023 (FIPA, 2002), which defines the FIPA Agent Management Specification.

Concepts

a) Concept **(metadata :content** *string*): This concept asserts that there is a data model which contains metadata information about the educational content of the intelligent learning object. This information is contained in the **:content** parameter, and must be compliant with the *IEEE 1484.12.1 Standard for Learning Object Metadata* and must be coded using the *IEEE 1484.12.3 Standard for Learning Technology – Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata*.

b) Concept **(dataModel :content** *string*)**:** This concept asserts that there is a data model which contains information about the interaction between a student and an ILO. This information is contained in the **:content** parameter, must be compliant with the *IEEE 1484.11.1 Standard for Learning Technology – Data Model for Content Object Communication*.

c) Concept **(learner :name** *string* **:id** *string* **:data-model** *string*): This concept asserts that there is a student with the name defined in **:name**, which has an unique identifier contained in **:id**. The information about the interaction between this student and a ILO is contained in **:data-model**. This information must be compliant with the *IEEE 1484.11.1 Standard for Learning Technology – Data Model for Content Object Communication*.

d) Concept **(ilo :agent-id** *string* **:metadata** *string* **:location** *string*): This concept asserts that there is the ILO which metadata information is defined in **:metadata**. If this agent is operating in the agent society in a certain moment, the :agent-id parameter has its unique identifier. In the other case, :location has a reference for the location where the agent is found (for example, the agent's java class). The :metadata parameter must be compliant with the *IEEE 1484.12.1 Standard for Learning Object Metadata* and must be encoded with the *IEEE 1484.12.3 Standard for Learning Technology – Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata*.

Actions

a) Action **(send-metadata):** Must be used when an agent needs the metadata information of ILO. This action does not have parameters

b) Action **(send-learner):** Must be used when an agent needs information about the student. This action does not have parameters.

c) Action **(search-ilo :metadata** *<metadata>*)**:** Must be used when an agent needs to have the ILOR sending information about ILOs that satisfy the criteria defined in the **:metadata** parameter. This parameter must contain the minimum set of features that the ILO must have to be part of the result set of the action.

d) Action **(get-learner-lms :learner** *string* **:ilo** *string*): Must be used when an agent needs to have the LMS sending information about the student **:learner** related to

the ILO **:ilo**. The parameters :learner and :ilo are unique identifiers for the student and the ILO, respectively.

e) Action **(put-learner-lms :learner** *<learner>* **:ilo** *<ilo>***):** This action must be used when an agent needs to have the LMS storing the information about a student **:learner** related to the ilo **:ilo**. The parameters :learner and :ilo contain the information about the student and the ILO.

f) Action **(put-learner-ilo :learner** *<learner>***):** This action must be used when an agent needs to have the ILO evaluating the information about a student **:learner**. The parameter :learner contain the information about the student.

g) Action **(activate :ilo** *<ilo>***):** Must be used when the ILO needs to have its status changed to activated in the ILOR's list of activated ILOs.

h) Action **(deactivate :ilo** *<ilo>***):** Must be used when the ILO needs to have its status changed to deactivated in the ILOR's list of activated ILOs.

3.4   Dialogues

Using the FIPA-Request protocol, we defined a set of conversations protocols that the agents must be able to perform. In this session we show some of them.

Registering in DF
The DF (Directory Facilitator) Agent is part of the FIPA Agent Reference Model. It works as a directory facilitator where agents can register the services they can provide to the other agents. The document SC0000023 (FIPA, 2004) specifies how this process must be made.

Requesting metadata information from an ILO
To request the metadata information of an ILO, agents must use the dialogue **get-metadata**. This dialogue is initiated by a request message containing the action **send-metadata** sent to the ILO. If the ILO agreed to execute the task when it is completed with success, the final message contains a **metadata** concept containing the ILO's metadata information. Table 1 shows the phases of this dialogue according to the FIPA-Request protocol.

Table 1. Phases of the get-metadata dialogue according to the FIPA-Request protocol.

Phase	Sender	Receiver	Performative	Content
1	Agent	ILO	**Request**	(**action** *<AID>* *<send-metadata>*)
2	ILO	Agent	**Agree**	(<phase 1 content>
	ILO	Agent	**Not-**	(<phase 1 content>
	ILO	Agent	**Refuse**	(<phase 1 content> <reasons>)
3	ILO	Agent	**Inform**	(**result** <phase 1
	ILO	Agent	**Failure**	(<phase 1 content> <reasons>)

Requesting learner information from an ILO

To request the information about the learner which the ILO is interacting with, agents must use the dialogue **get-learner**. This dialogue is initiated by a request message containing the action **send-learner** sent to the ILO. If the ILO agrees to execute the task and carries it out successfully, the final message contains a **leaner** concept containing the learner's information. Table 2 presents the phases of this dialogue according to the FIPA-Request protocol.

Table 2. Phases of the get-learner dialogue according to the FIPA-Request protocol.

Phase	Sender	Receiver	Performative	Content
1	Agent	ILO	**Request**	(**action** *<AID>* *<send-learner>*)
2	ILO	Agent	**Agree**	(<phase 1
	ILO	Agent	**Not-**	(<phase 1
	ILO	Agent	**Refuse**	(<phase 1 content> **<reasons>**)
3	ILO	Agent	**Inform**	(**result** <phase 1
	ILO	Agent	**Failure**	(<phase 1 content> **<reasons>**)

Requesting learner's information from an LMS Agent

The dialogue **get-learner-lms** must be used to request information about the interactions of a given student to the LMS agent. This dialogue is initiated by a request message, sent to a LMS Agent, containing the action **get-learner-lms.** This action contains the learner's identifier and the ILO's identifier. If the sender agent wants to obtain student/ILO information, the :ilo parameter must be filled. In the other case, the LMS Agent will send information about the learner and all the ILOS. The final message contains a set of **dataModel** concepts containing the requested information. Table 4 presents the phases of this dialogue according to the FIPA-Request protocol.

Table 3. Phases of the get-learner-lms dialogue according FIPA-Request Protocol.

Phase	Sender	Receiver	Performative	Content
1	Agent	LMS	**Request**	(**action** *<AID>* **<get-learner-lms>**)
2	LMS	Agent	**Agree**	(<phase 1 content>)
	LMS	Agent	**not-understood**	(<phase 1 content>)
	LMS	Agent	**Refuse**	(<phase 1 content> **<reasons>**)
3	LMS	Agent	**Inform**	(**result** <phase 1
	LMS	Agent	**Failure**	(<phase 1 content> **<reasons>**)
	ILO	Agent	**Failure**	(<phase 1 content> **<reasons>**)

3.3 Dynamic of the ILO society

The proposed agents in this work have three different states: not-initialized, activated e finalized.

In the first state, the agent is not instantiated or is not registered in the platform. When the agent is instantiated it must register itself in the DF and in the AMS using the dialogues defined by FIPA in the document SC0000023 (FIPA, 2002) and the schema for registering in those entities defined in this work. If the agent is an ILO, it must change its status within the ILOR Agent using the dialogue **modify-status** and the action **activate**. After this, the agent will be in the activated state. Then it is able to execute the tasks for what it has been defined for.

To get to the finalized state, the agent must deregister itself in the DF and AMS. To do so it must use the dialogues defined by FIPA in the document SC0000023 (FIPA, 2004) for this purpose. If the agent is an ILO, before deregistering in the DF and AMS, the agent must change its status within the ILOR Agent using the dialogue **modify-status** and the action **deactivate**. As soon as the agent gets the finalized state, it goes to the not-initialized state.

4 Conclusions

This paper presented the communication structure proposed for the ILO approach and how it can be used to improve interoperability among learning objects and pedagogical agents. At this point, our conclusion is in the way that we need to stop thinking of learning objects as chunks of instructional content and to start thinking of them as small, self-reliant computer programs. This means more than giving a learning object some sort of functionality, more than writing Java calculators or interactive animations. When we think of a learning object we need to think of it as a small computer program that is aware of and can interact with its environment. The agent approach can enable these features. Intelligent Learning Objects are able to improve the adaptability, interoperability and interactivity of learning environments built with these kinds of components by the interaction among the learning objects and between learning objects and other agents in a more robust conception of communication rather than a single method invocation as the object-oriented paradigm use to be.

5 References

Advanced Distributed Learning (ADL). (2004). *Sharable Content Object Reference Model (SCORM ®) 2004 Overview*. retrieved July 7, 2005 from www.adlnet.org.

ARIADNE (2004). *Alliance of remote instructional authoring & distribution networks for Europe* Retrieved July 7, 2005 http://ariadne.unil.ch.

Bradshaw, J. M. (1997). An introduction to software agents In: Bradshaw, J. M. Ed. *Software Agents*. Massachusetts: MIT Press, 1997.

CanCore (2004). Canadian Core About. Retrieved July 7, 2005 from http://www.cancore.ca/about.html.

Downes, S. (2001). Learning objects: resources for distance education worldwide. in *International Review of Research in Open and Distance Learning*, 2(1). 2001

Downes , S. (2002). *Smart Learning Objects*.

EMORPHIA (2005). *FIPA-OS - FIPA - Open Source*. retrieved September 19, 2005 from http://fipa-os.sourceforge.net/index.htm.

FIPA: The foundation for Intelligent Physical Agents (2002). *Specifications*. retrieved July 7, 2005 from http://www.fipa.org.

Friesen, Norm (2001). What are Educational Objects? Interactive Learning Environments, 3 (9).

IEEE Learning Technology Standards Committee (2004). *Specifications*. retrieved July 7, 2005 from http://ltsc.ieee.org.

IMS Global Learning Consortium. (2004). *Current specifications*. retrieved July 7, 2005 from http://www.imsglobal.org /specifications.cfm.

Mohan, P.and Brooks, C. (2003). Engineering a Future for Web-based Learning Objects. *Proceedings of International Conference on Web Engineering*, Oviedo, Asturias, Spain.

Silveira, R. A., Gomes, E. R, Vicari, R. M. (2005). Inteligent Learning Objects: An Agent-Based Approach of Learning Objects. In Weert, Tom Van, Tatnall, Arthur (Eds.) *Information and Communication Technologies and Real-Life Learning*.Boston Springer, 1103 - 110.

Wooldridge, M.; Jennings, N. R.; Kinny, D (1999). A methodology for agent-oriented analysis and design. In: Proceedings of International Conference on Autonomous Agent AAMAS, v.3. 1999.

Acknowledgment