

ILOMAS: An Intelligent Learning Objects Implementation Study Case

João de Amorim Jr. and Ricardo Azambuja Silveira^(✉)

PPGCC – UFSC, Florianópolis, Brazil
joao.amorim.jr@gmail.com,
ricardo.silveira@ufsc.br

Abstract. This paper presents the implementation and evaluation of ILOMAS, an architectural model designed to select Learning Objects (LO) for e-learning, based on multi-agent systems. The proposed model extends the Intelligent Learning Objects approach through the use of a BDI agent architecture, allowing the communication with the instructional resources that constitute the LO according to the SCORM standard. A prototype implementation is presented to evaluate the proposed model.

Keywords: Dynamic learning experience · LO · LMS · BDI · Multi-agent system

1 Introduction

The Intelligent Learning Objects Multi-Agent Systems (ILOMAS) is a dynamic virtual learning environment, based on the Multi-agent systems (MAS) approach, integrated to a Learning Management System (LMS), which aims to include Learning Objects (LO) dynamically, adding intelligent behavior to LMS. The proposed model improves reusability by the combination of LOs available in some LO repositories and adaptability by selecting new Learning Objects from the repository according to the student's performance and behavior. This paper presents the implementation details of the ILOMAS prototype. The detailed description of the conceptual model, the design of the proposed architecture, the theoretical background and a discussion of related works was previously presented in [11, 12].

To produce more intelligent LO, previous works [2, 3] proposed the convergence between the LO and MAS technologies, called Intelligent Learning Objects (ILO). This approach makes possible to offer more adaptive, reusable and complete learning experiences. An ILO is an agent capable to play the role of a LO, which can acquire new knowledge by the interaction with students and other ILO (agents' information exchange), raising the potential of student's understanding.

This new version of the model called Intelligent Learning Object Multi-Agent System (ILOMAS) is composed by agents with specific goals, and capable of communicating and offering learning experiences to students in an LMS course, according to the interaction with these students.

2 Design and Implementation

The ILOMAS framework was implemented according to the architecture proposed by De Amorin and Silveira [10, 11] as a prototype aiming to validate the proposed model. The JADEX framework [5, 6] was chosen to implement the agents based on the BDI architecture [1, 4]. The JADEX platform supports the creation of active components which is an approach that gets benefits from the association of two distinct technologies: Agents e SCA. The SCA model was proposed by IT companies (i.e.: IBM, ORACLE) with the intention of promoting the interoperability among distributed applications, according to concepts of components and service oriented architecture (SOA) [5, 6].

The interaction interface between the student and the agents' environment was implemented based on the Java Servlets and JSP technologies, getting benefits of the JADEX BDI V3 services communication structure. The Servlets technology allows the execution of services and Java classes at the server side from Web requests.

On the prototype, the servlet layer delegates the handle of the student browser request to a Java class (non-agent) based on the Facade design pattern. This pattern provides a unified and simplified interface to a sub-system, promoting low coupling [7]. The ILOMASFacade class offers to the servlet classes and the access to agents' services (the agents' capabilities, plans, etc.) keeping the separation between the MAS layer and the external items (front-end and servlets), avoiding unnecessary coupling (Fig. 1) [8].

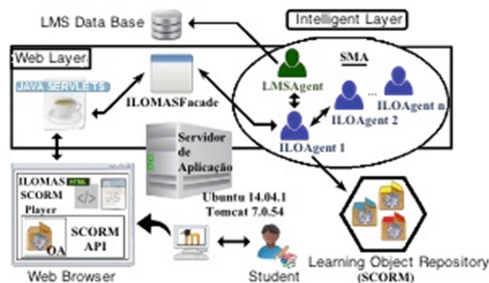


Fig. 1. ILOMAS web prototype architecture

3 Evaluation

Two different prototypes of the proposed model were implemented. The very first prototype was deployed to run over the Apache Tomcat Server (7.0.57) to perform the first simple tests of the proposed model. In this version of the prototype, it was built a scenario which, while the system runs, a student accesses the system and the LMSAgent identify that the student needs to study Photosynthesis. After that, the student ask for the learning experience leading to a new servlet requesting. This servlet forward it to ILOMASFacade, which waits for the ILOAgentBDI's deliberation. Then, this agent identifies some LO related to this subject matter within the repository. Finally, this LO was shown to the student successfully.

Besides, during this test it was simulated a student's request for a complementary learning experience (by pressing the corresponding button). As result, a new ILOAgentBDI instance was created in the system, which searched for and found a different LO related to the same subject (photosynthesis) within the repository. It was not explicitly defined in the database that the student should have watched this new LO (only the subject was required, any specific LO), so the MAS obtained the related LO dynamically by itself.

Further, it was simulated a student's request for a complementary learning experience (by pressing the corresponding button). As result, a new ILOAgentBDI instance was created on the system, which searched for and found a different LO related to the same subject (photosynthesis) within the repository. It was not explicitly defined in the database that the student should have watched this new specific LO (only the subject was required, no specific LO), so the MAS obtained the related LO dynamically.

3.1 ILOIR: Moodle Plug-in

The next step to evaluate the proposed model was, to integrate the environment ILOMAS with the LMS Moodle in a transparent manner, so that the student does not need to leave the virtual environment neither access any other system, preventing possible loss of attention. We developed a Moodle plug-in as an interface resource to the Intelligent Learning Objects, called ILOIR (Intelligent Learning Object's Resource interface). This resource allows the teacher to define: the subject matter (theme) and the learning objective of the instructional unit; and the address where the ILOMAS platform is available (Fig. 2). Furthermore, the ILOIR plug-in provides access to the functionality of ILOMAS system for the student, within the Moodle. The developed extension engages the ILOMAS the web layer (SCORM player, container HTML and JavaScript library) in the LMS structure, allowing the access to these features in Moodle and establishing (indirect) communication to the learning environment with the agents.

The screenshot shows a Moodle page titled "Atualizando um ILOIR em Tópico 1". The page is divided into sections: "Geral" (General), "Nota" (Note), and "Configurações comuns de módulos" (Common module configurations). The "Geral" section contains a form with the following fields:

- Nome*** (Name): Introdução ao Direito Previdenciário
- Descrição** (Description): Objeto Inteligente de aprendizagem referente ao assunto da Unidade 3
- Assunto*** (Subject): Introdução ao Direito Previdenciário - Regimes Próprios
- URL da plataforma ILOMAS (Acesso aos Objetos de Aprendizagem)*** (ILOMAS platform URL): http://posiate.inf.ufsc.br:8080/ILOMASWeb

The "Nota" section is currently empty. The "Configurações comuns de módulos" section is also empty.

Fig. 2. ILOIR screenshot



The screenshot shows the Moodle interface for a course titled "Direito Previdenciário". The language is set to "Português - Brasil (pt_br)" and the user is "Professor 1". The breadcrumb trail is "Página inicial > Meus cursos > Direito > DirPrevSCORM". A "Ativar edição" button is visible. The course content is organized into three topics, each with a "Fórum de Avisos" link and a document icon representing the course material:

- Tópico 1**: Introdução ao Direito Previdenciário - Unidade 1
- Tópico 2**: Introdução ao Direito Previdenciário - Unidade 2
- Tópico 3**: Introdução ao Direito Previdenciário - Unidade 3

Below the topics is a "Material Complementar" section with a document icon and the text "Sistema de Legislação da Previdência Social". A vertical navigation menu on the left includes "Navegação", "Administração", "Fóruns", "Notícias", and "Eventos".

Fig. 3. Structure of the course with Scorm



The screenshot shows the Moodle interface for the same course "Direito Previdenciário". The language is "Português - Brasil (pt_br)" and the user is "Professor 1". The breadcrumb trail is "Página inicial > Meus cursos > Direito > DirPrevILOIR". A "Ativar edição" button is visible. The course content is organized into one topic and a complementary material section:

- Tópico 1**: Introdução ao Direito Previdenciário

Below the topic is a "Material Complementar" section with a document icon and the text "Sistema de Legislação da Previdência Social". A vertical navigation menu on the left includes "Navegação", "Administração", and "Fóruns".

Fig. 4. Structure of the course with ILOIR

3.2 Experiments

The second experiment to evaluate the proposed architecture is to use the model for the creation of an introductory course in Social Security Law, offered by a school of government of a public agency statewide. The evaluation model aimed to compare two types of scenarios: one is a course set normally in a conventional way and another one set dynamically according to the student interaction using the ILOIR features. For that, we performed some comparisons between two courses created in Moodle (Fig. 3 and 4).

The first course was built with three LOs about Social Security Law (available in the repository) included as SCORM objects resources (native in Moodle). The second course was set just including only one ILOIR type resource (usin the ILOIR new developed plug-in), whose subject refers to the subject of the first SCORM object inserted in the course mentioned above. In order to facilitate the description, hereinafter the course with SCORM objects will be called “Course 1” and the course with ILOIR feature is called “Course 2”.

Table 1. LMS configuration course

	Included resources	Included OAs	Time to configure	Total time to configure	Available OAs
Course 1	3	3	00:02:18	00:08:30	1
Course 2	1	0	00:00:58	00:04:30	Total amount of OAS available on the repository

Tables 1 and 2 compare the two mentioned courses set in Moodle. While the course 1 uses the explicitly included LOs, the course 2 presented several different OA to the student during the experiment, related to the object matter theme set by the teacher using the ILOIR feature. This feature offered by the plug-in facilitates the reuse of OA. It is important point to observe that only one object was included by the teacher in the course 2.

Table 2. ILOIR students interaction

Student	Partial concept	Number of subjects demonstrated	Student’s attention time	Number of wrong answers	Extra OAs (reinforcement)
1	Good	1	00:09:46	0	0
2	Regular	2	00:21:13	3	1
3	Bad	3	00:34:19	6	3

With regard to student interaction scenarios with the plug-in ILOIR, we evaluated the access of three students with different degrees of understanding of Social Security Law (concepts: poor, fair and good), according to results of Table 2.

To note that, depending on the student’s performance (the amount of errors in the assessment test in OA, or the delay in interacting with the object), additional new LOs are displayed to the student the same learning session. Is therefore the adaptation of dynamically OA content, as resolved by the agents (intelligent system layer), according to student performance (Figs. 5 and 6).

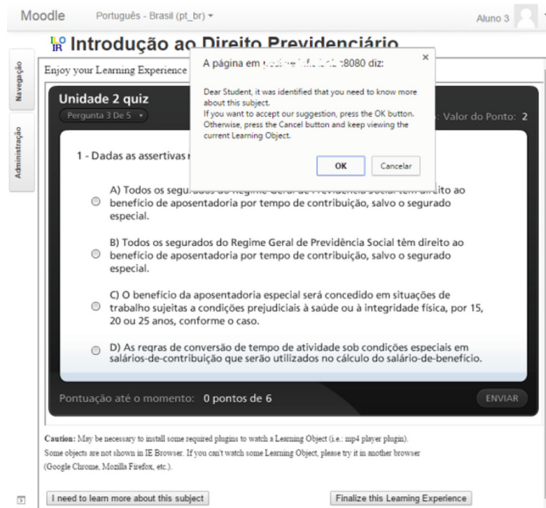


Fig. 5. Diagnosis and signaling new LO to the student 3, during OA display related to Unit2

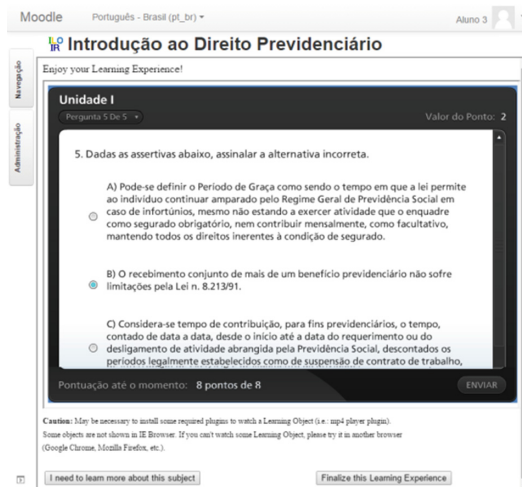


Fig. 6. Provision of prerequisite LA (Unit1).

4 Conclusions and Future Work

This paper presented the prototype implemented to evaluate a model to build more adaptive and reusable educational experiences, using learning objects. The ILOMAS framework was designed to allow the dynamic LO selection on LMS courses, as an improvement of ILO's previous approach. The agents are modeled based on the practical reasoning paradigm (towards goal achievement). The MAS was developed

following the JADDEX BDI V3 framework, which permits that the agents' functionalities can be accessed as services. The use of Servlet technology provides the integration of front-end and intelligent layer.

The prototype was implemented to verify the proposed model, and some evaluation tests were executed. As result, the ILOMAS has received the learning experience requested by the student, and has identified dynamically a LO associated with the subject that the student must have learned about (according to the LMS database information).

We are aware that this is just very small observation of the system's performance and the validation performed is still small to affirm the success of the proposed model. New tests will be done with the system with more objects and more students. As future work, the ILOMAS framework will be extended to supply the integration with LO recommendation systems based on ontologies in order to raise reuse, dynamic sequencing, and interoperability. Moreover, the model will be tested with different learning situations and real students.

References

1. Wooldridge, M.: An Introduction to MultiAgent Systems, 2nd edn. Wiley, Chichester (2009)
2. Silveira, R., Gomes, E., Vicari, R.: Intelligent Learning Objects: An Agent-Based Approach of Learning Objects. In: IFIP – International Federation for Information Processing, vol. 182, pp. 103–110. Springer-Verlag (2006)
3. Silva, J., Silveira, R.: The development of intelligent learning objects with an ontology based on SCORM standard. In: Seventh International Conference on Intelligent Systems Design and Applications, pp. 211–216. IEEE (2007)
4. Pokahr, A., Braubach, L., Haubeck, C., Ladiges, J.: Programming BDI agents with pure Java. In: Müller, J.P., Weyrich, M., Bazzan, A.L. (eds.) MATES 2014. LNCS, vol. 8732, pp. 216–233. Springer, Heidelberg (2014)
5. Pokahr, A., Braubach, L., Jander, K.: The Jadex project: programming model. In: Distributed Systems and Information Systems, Chap. 1, pp. 1–34. University of Hamburg (2012)
6. JADDEX. <http://www.activecomponents.org/bin/view/About/Features>
7. Prometheus Design Tool (Eclipse Plug-in). <https://code.google.com/p/pdt-plugin/>
8. Braubach, L., Pokahr, A.: Jadex Active Components Framework – BDI Agents for Disaster Rescue Coordination. University of Hamburg (2011)
9. Braubach, L., Pokahr, A.: Developing Distributed Systems with Active Components and JADDEX. University of Hamburg (2012)
10. SCORM 2004. Advanced Distributed Learning. <http://www.adlnet.org/scorm>
11. de Amorim Jr., J., Silveira, R.A.: Dynamic selection of learning objects based on SCORM communication. In: Pereira, F., Machado, P., Costa, E., Cardoso, A. (eds.) EPIA 2015. LNCS, vol. 9273, pp. 658–663. Springer, Heidelberg (2015)
12. de Amorim Jr., J., Gelaim, T.Á., Silveira, R.A.: Dynamic e-learning content selection with BDI agents. In: Bajo, J., Hallenborg, K., Pawlewski, P., Botti, V., Sánchez-Pi, N., Duque Méndez, N.D., Lopes, F., Vicente, J. (eds.) PAAMS 2015 Workshops. CCIS, vol. 524, pp. 299–308. Springer, Heidelberg (2015)