

Pedagogic Design Assistant

Regina Motz
Instituto de
Computación, Facultad
de Ingeniería,
Universidad de la
República, Uruguay
[reginamotz,nderogatis,
nicolasmillot,javoliva9
8]@gmail.com

Natalia De Rogatis
Instituto de
Computación, Facultad
de Ingeniería,
Universidad de la
República, Uruguay
[reginamotz,nderogatis,
nicolasmillot,javoliva98
]@gmail.com

Nicolás Millot
Instituto de
Computación, Facultad
de Ingeniería,
Universidad de la
República, Uruguay
[reginamotz,nderogatis,
nicolasmillot,javoliva98
]@gmail.com

Javier Oliva
Instituto de
Computación, Facultad
de Ingeniería,
Universidad de la
República, Uruguay
[reginamotz,nderogatis,
nicolasmillot,javoliva98
]@gmail.com

ABSTRACT

Nowadays, most academic courses have a web site associated with information about them; however they are not based on a formal pedagogic design. It is used to include on the web site all the material used on a traditional course online, without modifying its sequence or the way the information is presented, to a blended or distance course. The Pedagogic Assistant developed propose a solution to this problem, using SCORM and an extended LOM as standards, as well as a customize rules mechanism that allows the formalization of a pedagogic design which will be used to validate no just blended or distance courses, but also any digital course material.

KEYWORDS

Pedagogic Design Evaluation

INTRODUCTION

In recent years, there has been a huge activity in the development of web-based courses. Building materials for these courses has been shown to be the most costly in terms of both time and money, for what quickly became necessary, the reuse of these materials. The contents are then represented in the form of *learning objects*. A learning object is defined by the IEEE as "any object, digital or not, which can be used, reused, or referenced in the teaching process supported by technology", and according to Wiley [14] as "any digital resource that can be used repeatedly to facilitate learning". The above definitions agree that the learning objects must be reusable. Moreover, in addition to this feature of reusability, learning objects are commonly evaluated from the point of view of its educational content. Some initiatives in this direction include the Learning Object Instrument Rating (LORI) [9], and MERLOT [6]. However, to the best of our knowledge, there is any tool where learning objects are

automatically evaluated in relation to their suitability for a pedagogical design.

Our main concern is to develop a tool that assists the teacher to produce e-courses following the guides of a pedagogical model. Our focus is on the belief that the content of a course is not contained in its parts, but that emerges from the combination of the parties. In order to achieve this challenge learning objects (LO) need to be annotated by domain tailored metadata, such as Learning Object Metadata, LOM [7]. But normally, LO do not have sufficient metadata. This is natural since the manual generation of metadata is an expensive task. One way to reach this goal is to use Learning Content Management Systems (LCMS), such as Atutor, Moodle or Claroline. One of the characteristics of LCMS is that they allow authors to create course materials with the possibility to be exported using standards like LOM. However, when authors already have previous materials, like e.g. course's web pages, it is not easy to incorporate them to the LCMS with clear semantics. Instead of LCMS, there are several technologies that permit reusability, shareability and interoperability among web applications. Ontologies and web standards, such as XML, RDF and OWL, allow the specification of components in a standard way. Our proposed Pedagogical Assistant use these technologies to automatically markup course web pages based on their content and to produce standard packages of learning objects using SCORM (Sharable Content Object Reference Model [13]) an ADL initiative [1]. There are also other tools that provide the possibility to automatic annotated LO.

In this work we present a tool that given an e-course implemented by LO annotated with LOM and packaged with SCORM, is able to automatically evaluate the affinity of its pedagogical design with respect to a given pedagogical model.

The current pedagogical models have analyzed the contemporary reality of educational services, focusing on

Motz, R., De Rogatis, N., Millot, N., Oliva, J. (2008). Pedagogic Design Assistant. En J. Sánchez (Ed.): Nuevas Ideas en Informática Educativa, Volumen 4, pp. 155-160, Santiago de Chile.

different aspects, such as the relationship between teachers and students, content and the transmission of culture and its interaction with society. Thus we find very different teaching practices that can be analyzed as ways to design and act in education, and making up what we call pedagogical models. In addition, these models are correlative in the pedagogical design, defined in ways to develop the interactions between teachers and students and among the students themselves. From our pedagogic model approach becomes essential to make clear the kind of interaction between teacher and student and the relationship both with the content, as well as among the students themselves. Then, e-courses must include activities that promote the availability of exercises and the conceptual integration of concepts, individual practical activities and also the interactions such as forums and chats.

The rest of the paper is organized as follows. First we describe the basic concepts of learning object metadata and SCORM. Then we present the methods and techniques we use to develop a Pedagogic Assistant, and we depict the representation of the pedagogical model. Following this we present a case of study. Finally, we bring some conclusions and describe future work.

LEARNING OBJECT METADATA

A Learning Object is a learning unit that can be re-used and sequenced in order to build a more complete course. These Learning Objects may contain metadata that adds information to the learning unit. Institutions like the IEEE and IMS have been working in the specification of metadata for learning objects. LOM [7] is a widely accepted metadata standard.

The LOM metadata is organized on 9 categories: General, Life Cycle, Meta-Metadata, Technical Aspects, Educational Aspects, Copyright, Related Resources, Annotations, Classification. From the Educational category we use the following fields: "Learning Resource Type", "Interactivity Type" and "Interactivity Level", described in Table 1. Although the LOM specification has many fields, we decided on focusing on the mentioned before, since these contain the main information to be able to analyze the pedagogical design.

LOM Field	Values
Learning Resource Type	exercise, simulation, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, lecture, problem statement, self assessment, questionnaire
Interactivity Type	active, expositive, mixed
Interactivity Level	very low, low, medium, high, very high

Table 1. Fields from LOM Educational Category.

However, these fields did not have enough coverage; in particular, there are insufficient values at the Learning Resource Type field, as the types *example* or *definition*. We decide to use then, the Ullrich's LOM extension [3 ,4] depicted in Table 2.

LOM Field	Extension Values
Learning Resource Type	example, definition, notice board, time-table, program, scheduler, automatic receiver

Table 2. LOM Learning Resource Type extension [4].

The ADL initiative SCORM (Shareable Content Object Reference Model) [13] is a widely adopted standard that introduces a complex idea, called *sequencing*, which are rules that specify the order in which a learner may experience content objects. SCORM has been adopted by numerous e-learning platforms as the standard way of import/export course contents

PEDAGOGIC ASSISTANT APPROACH

In this section we present our Pedagogic Design Assistant. The aim of this project is to assists teachers to evaluate their e-course's design according to a given pedagogical model. Figure 1 shows the overall architecture of the Pedagogical Assistant Project [11]. The whole process is implemented as client-server architecture with a web graphic interface; all the application logic is centralized on the server.

The LOM model of the course is implemented as a Course Ontology in OWL [10]. This ontology is used to guide the topic learning objects and the LOM markup. It describes the elements of the course for each specific university or area. For example, in different courses the evaluation method could be different: exam, practical tasks, writing jobs, research works, etc. For a given Pedagogical Model the Assistant will decide if the e-course satisfied all its definitions w.r.t. LO or if some extra activities must be added or removed from the e-course.

A pedagogue specifies the rules that will determine the pedagogic design, this is done using a business rules language. He/she also has to define the notifications that the Pedagogical Assistant gives to the teacher. These rules and notifications define the pedagogic design, and are introduced at the implementation face of the tool. After this step the tool can be used by any teacher that has a SCORM package for a course. The teacher has to prepare the SCORM package from the course, using an external tool (i.e. Reload Editor [12]), identifying the course Learning Objects and its metadata. The result of the Pedagogic Assistant will depend on how detailed is the information the teacher adds to the SCORM package. This SCORM package must be an input for the tool.

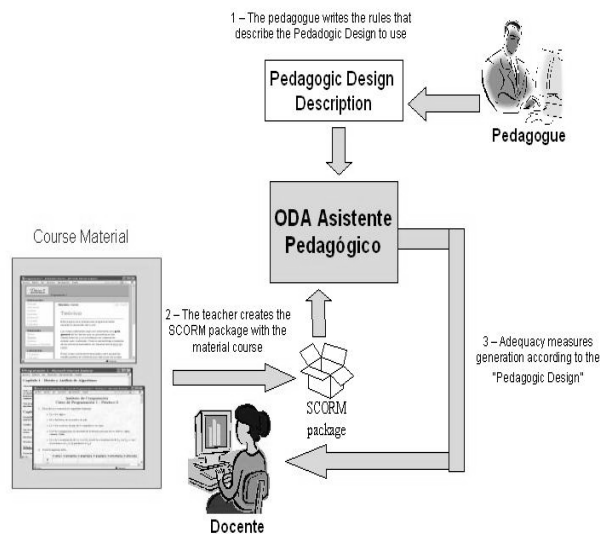


Figure 1. Pedagogic Assistant Architecture.

Once the SCORM package has been prepared, the user (in this case the teacher) enters the path to the package location on the Pedagogic Assistant. Then the tool decompresses the package and analyzes it. After running the tool, a set of notifications is displayed to the user on the web browser. Later on, if the user chooses to follow the suggestions made by the Pedagogic Assistant, he/she could edit the SCORM package on an external tool and apply the changes suggested by it. Then, the Pedagogic Assistant may be applied again on the edited SCORM package. This SCORM package may be uploaded to an educational platform with SCORM compliance, like Atutor [2] or Moodle [8]. Furthermore, the user may comment about the notifications displayed by the tool. These comments are stored along with the notification, indicating whether the user agrees with it or not.

The Pedagogic Assistant also provides other functionalities apart from the notifications. It provides reading time estimations for each resource included on the package (at this point only the following formats are estimated: pdf, txt, html/htm); as well as statistics that show an historic of the tool use.

Relevant Aspects	jProlog	Jess	Drools
Forward Chaining	No	Yes	Yes
Easily integrated with Java	Yes	Yes	Yes
Free	Yes	No	Yes
Intuitive syntax rules	No	Yes	Yes

Tabla 2. Comparing Different Rules Engines.

PEDAGOGIC MODEL

The representation of the Pedagogic Model was supposed to be done in a way that it can be easily modify by people who do not necessarily have advanced computer knowledge. We choose for this representation the use of Business Rules since we needed to specify certain behavior based on the data available. The question that arises is: What Type of Business Rules we need? We need business rules like "If X we do Y", best known as "forward chaining" and that could be evaluated in a dynamic way.

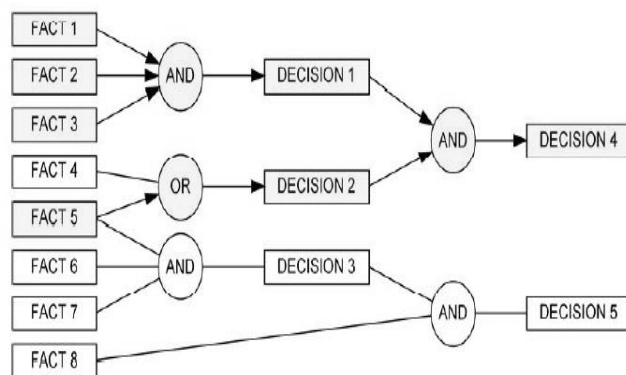


Figure 2: Forward Chaining.

As we use such rules, we were making a guided system by the data available to us. Rules of this kind, could be considered related to a sentence IF/THEN in a programming language.

The main idea here is to centrally manage the statements of the business rules and determine whether the conditions of some of them are true. If this is the case, then see what action should be taken.

In our case, when the rules are true, information is sent to a class that centralizes these messages, and that class is the one that decides what information will be showed to the end user. We use these rules not to change the status or value of any object, but simply serve to generate notifications and store this data in the database.

Table 2 shows a comparison between some different Rules Engines. The Engine Rules have an advantage which is the promotion of a declarative approach, they say what should be done but without saying how to do that, in addition they centralized the knowledge and are easily understood by domain experts, things that apply perfectly to our needs. From Table 2 and *forward chaining* presented before, we came to the decision to use Drools as an engine of our Business Rules Project.

DROOLS

Drools [5] (also known as JBoss Rules) is a Business Rules engine implemented entirely in Java, free and chained forward. The objects that are created are stored in a working memory, and these objects are then evaluated to determine what rules meet their conditions and which do not. The rules, which assess conditions in True, cause to run the instructions that were declared in their respective consequences.

In the format of the statement of the rules we can distinguish three distinct parts:

- **Head:** Defines the name of the rule and the properties of it (such as the number of priority, etc.).
- **Conditions:** Data or functions of Java classes whose values are to be checked.
- **Consequences:** Actions to be taken if the rule is triggered.

```
rule "<rule name>" <attribute> <value> (Head)
when <Java class> (<attribute>,
<functions>) (Conditions)
then <instruction1>...<instructionN>
(Consequences)
end
```

The conditions of a rule can be defined based on the values of the attributes from one or more Java classes and evaluations of their respective roles. Thus, creating instances of Java classes, inserting them in the working memory and firing the evaluation of the rules, determines the rules that are fired at a particular time.

The statement of the rules is centralized in a single flat file extension, `drl`, (in our case, the file is called `Reglas.drl` and is within the source code delivered in the project home page <http://www.fing.edu.uy/~rmoz/PedagogicAssistant.htm>). The parser is responsible for checking the correctness of these syntactic statements and generates an intermediate structure that then becomes in packages that form the basis of the knowledge.

The "Working Memory" is the main class that contains all the objects inserted as knowledge, and is where the true interaction with our application happens. After all the objects were added in the working memory, we fired them (with the command `fireRules`) so the rules are evaluated. Once evaluated, those with their conditions on True, execute the instructions declared in the consequences.

We present an example of a rule implemented by us:

```
rule "Look for a graphic"
salience 10
when o: ODA() eval
(o.getLomMetadata().searchEducationalLearnin
gResourceType("figure")) ||

o.getLomMetadata().searchEducationalLearnin
```

```
gResourceType("slide")) ||

o.getLomMetadata().searchEducationalLearnin
gResourceType("graph")) ||

o.getLomMetadata().searchEducationalLearnin
gResourceType("diagram"))
then String s = "At least it has a graphic";
System.out.println(s);
MessageHandler m =
MessageHandler.getInstance();
m.addRuleFired(9);

end
```

As shown in this example, we exploited the structure defined by LOM and through it, we conducted a search on each instance of ODA (Learning Object) to identify which have an element of the type graph. The syntax is very simple and it allows to do rules that otherwise would be converted into a code complex and extensive.

In addition to the rules, Drools gives us the option to handle queries (Query). The objective is to be able to perform certain queries (filters) on the objects we hold in our working memory. We show the format of the statement of a query:

```
query "<rule name >" (Head)
<Java class> (<attributes>, <functions>)
(Conditions)
end
```

A clear and simple example is:

```
query "Search simulation type"
o: ODA() eval
(o.getLomMetadata().searchEducationalLearnin
gResourceType("simulation"))
end
```

In this query we are looking for "simulation", again using the structure of LOM faithfully represented by the Pedagogic Assistant.

Examples Rules

In this section we briefly introduce some rules that we use in the Pedagogic Assistant. The major focus of our assistant is the field Learning Resource Type that is within the field Educational of LOM. We group the field Learning Resource Type in two classes:

- **Graphic (Gráfico):** diagram, graph, slide, figure.
- **Text (Texto):** exercise, simulation, questionnaire, index, table, narrative text, exam, experiment, problem statement, self assessment, lecture, definition, example, FAQ, theorem, activity, conclusion, demonstration, goal, midterm examination.

Furthermore, we add other types that are meaningful to the organization of the course:

- notice board (*cartelera*)
- time-table (*horarios*)
- program (*programa*)
- scheduler (*cronograma*)
- automatic receiver (*entrega de tareas online*)

For the mails, forum and newsgroup, we used the VCARD standard. In addition, we made a system of weighted average for evaluating the interactivity of a course, which assigned values to each type of resource. Then we grouped the rules defined in a group system based on the area involved in every rule.

For example:

- The group A makes reference to the index of the course or of the sections.
- The group K makes reference to the graphic that are fundamental for the participatory model we represent.
- The group L emerge from the ontology “Learning Styles” from th project Red Educa, it specificies the main features of learning styles Holistic Graphic, Analytical Graphic, Holistic Verbal, Analytical Verbal.

As another example, we transcribe in Table 3 some of the rules involving Groups A, K and L.

Group	Metadata to use	Description of the rule
A	index	Rates positive the existence of an index in the course, and it is better if it appears one for each section.
K	graph	Rates positive the existence of images on the course
L	Grafico, texto	The quantity of graphics surpassed the text. This is assessed as a percentage of the total of the ODA's of the course.

Table 3. Rules involving Groups A,K and L.

Each group has a notification system that reacts upon the activation or in the absence of activation of the rules of each group. An example for the Group K is depicted in Table 4.

Group	Notification comment
K	Emphasizes that the graphics are fundamental to our model. In case of a good proportion, it generates notification (26). If it appears in a small proportion generates notification (27).

Table 4. An example of Notification for Group K..

As an example, the following table shows the notification referred before.

Notification	In case that ...	Notification comment
26	graphics appear in good proportion	It presents a reasonable amount of graphics, which keeps the attention of the student.
27	graphics do not appear in good proportion	The images kept the attention of the student, it is recommended to use them based on the participatory model.

The Pedagogic Assistant also determines from the amount of rules that are fired of the L group, the learning style closer to the course.

CASE OF USE

The introductory course on programming, called “Programación 1”, has an associated web site with electronic material that complements the lectures. This web site contains different electronic materials, such as slides, exercises, course schedule, previous exams and their solutions, a link to the course newsgroup and an email address.

The material contained in the web page can be parsed to identify the Learning Objects and their metadata in LOM [7] standard. Doing this, we will have a package formed by the course electronic material, the corresponding Learning Objects and their metadata. This package is the input of our Pedagogic Assistant.

In particular, the course of “Programming 1” from the Instituto de Computación, UdelaR, contains 71 Learning Objects with the following types: diagram, figure, definition, example, exercise, schedule, index, newsgroup, and email. After running the Pedagogic Assistant to this particular course we observed that its level of interactivity is low, since the number of high interactivity Learning Objects is poor with the low interactivity ones; it does not contain objectives or goals in any section which is not a good indicator; the number of images (graph, diagram, figure, slide) is reasonable; it does not has exercises; it has a good connection between text and slides-graphics; it has a low level of activities; it has two ways of contact: mail and news and finally it points to a Holistic Verbal student.

CONCLUSIONS AND FUTURE WORK

In this paper, we describe our experience of implementing an application, called *Pedagogic Design Assistant*, that allows a teacher to receive some evaluation from an e-course with regard to a pedagogic model. The interaction between the Pedagogic Assistant and the user is through a simple and user-friendly interface, in which a user can access to the utilities of the Assistant through a user name and password, which handle different types of user (administrator or

teacher). The application sends notifications to the teacher about the evaluation of the course. These notifications can be used to pay attention to potential omissions in the design of the course (such as lack of examples and exercises), and also to indicate the degree of interactivity that introduced the course, suggestions on how to improve interactivity and also providing for an estimate for what kind of student is best designed the course (i.e. holistic or analytical) and also estimating the time required to read all the e-course material.

Moreover, the application obtains at the same time of the above explained notifications, teacher's assessment about the adequacy of the notifications. In this way, the user administrators can use them to improve the set of rules. And having information entered by other users is a lot of wealth for teachers who use the tool. Besides this, having statistical reports for use of the Assistant, helps users to find out if they are well focused in the use of it, be able to evaluate whether it is useful to them or not and for users administrators, knowing how well it is used the same by the users and whether their operation actually is expected or not

The information about the estimated times reading of the various resources of the course assessed by the assistant can be helpful for the teacher who uses the Wizard.

The fact that we have a tool developed as a client - server architecture, brings the advantage that it can be accessed from anywhere within a LAN, and that the use of it, is independent of the environment in which the final users (teachers, pedagogues, etc) are using the Assistant, such as the operating system, the user's PC, its browser, etc. We believe that the most important feature of the assistant is its extensibility, fundamental requirement raised from the outset, which becomes a key element for the success of future work.

Handling of several Pedagogical Designs, to the Assistant can be used by users of different and varied areas of education. Dealing with different profiles of users in order to differentiate them within a pedagogical design, as it is more than proved, because different studies on the matter, the existence of a wide variety of learning styles within a single pedagogical design.

Handling System Management Users graphic form (today is simple but must be done at the database level, without any impact on the rest of the tables or in the source). It should put a button on the login to give the option to create a new user on the system, provided it is done by user administrator.

Due to the use of Drools as Engine Rules of Business, has the possibility of having an interface for editing the rules in Natural Language through DSL (Domain Specific Language), which today is at an early version development for English.

The actual prototype can be found at <http://www.fing.edu.uy/~rmoz/JARDIN.htm>

Future work is the development of a Management System Notifications Web Interface, to change the descriptions of notifications, and at the same time to remove and create new notifications. Also, it will be very useful the inclusion of a semi-automatic system to generate the SCORM Packages, in order to create packages easily with courses electronic materials, and then turn it to use the wizard.

At this moment we are focus on the extension of the notification tool to be able to perform also recommendations of Learning Objects.

Acknowledgments

This work is partially supported by The Latin American and Caribbean Collaborative ICT Research and Microsoft under grant JARDIN-LACCIR RFP2007.

REFERENCES

- [1] ADL – Advanced Distribute Learning <http://www-adlnet.gov/>
- [2] ATutor Project. <http://www.atutor.it/login.php>
- [3] Carsten Ullrich, “Instructional Objects” Ontology. <http://www.ags.unisb.de/~cullrich/oio/InstructionalObjects.owl>
- [4] Carsten Ullrich. The learning-resource-type is dead, long live the learning- resource-type! Learning Objects and Learning Designs, 1(1):7-15, 2005.
- [5] Drools - <http://labs.jboss.com/drools/>
- [6] LeLoup Jean W, Ponterio Robert ON THE NET MERLOT: Multimedia Educational Resource for Learning and Onlin Teaching. Language Learning & Technology ,January 2002, Vol. 6, Num. 1, pp. 3-5 <http://llet.msu.edu/vol6num1/onthenet/>
- [7] LOM Learning Technology Standards Committee – http://www.iaa.upf.es/~jblat/material/doctorat/students/jccbis/Estandares_LOM.htm
- [8] Moodle Project <http://moodle.org>
- [9] Nesbit, J. C., Belfer, K. & Leacock, T. (2003). Learning object review instrument (LORI). E-Learning Research and Assessment Network.
- [10] OWL Editor Protégé - <http://protege.stanford.edu/>
- [11] Pedagogic Design Assistant Project <http://www.fing.edu.uy/~rmoz/PedagogicAssistant.htm>
- [12] RELOAD Project. <http://www.reload.ac.uk/>
- [13] SCORM – Shareable Content Object Reference Model. <http://www.adlnet.gov/scorm/index.cfm>
- [14] Wiley, D. A. (2002). The instructional use of learning objects. Bloomington, IN: Association for Educational Communications and Technology.