Architecture for Including Simulations in E-learning Systems: an Extension to SCORM

Angélica de Antonio Jiménez, Jaime Ramírez Rodríguez, Zayra Madrigal Alfaro
Facultad de Informática – Universidad Politécnica de Madrid
{angelica, jramirez@fi.upm.es}, z.madrigal@alumnos.upm.es

Abstract
The integration of interaction and simulation in e-learning systems represents a milestone in educational research and supports the student’s learning process in innumerable ways. Nevertheless, current standards do not provide appropriate mechanisms to treat simulations as learning objects, which makes their integration into e-learning systems a hard task. This work proposes an architecture as extension to SCORM which includes a Tutoring Module for Simulations (TMS). The main objective consists in providing mechanisms to track and “observe” the student’s actions while interacting with a simulation, thus enabling the TMS to take decisions or intervene when necessary, and/or to modify the simulation course.

1. Introduction
The constant development of information technologies has influenced enormously the current society. E-learning technology has caused important changes in traditional teaching-learning methods, and moreover, the application of simulations in this kind of educational environment can be considered as a great step in education and the enterprise world [1].

We can define the term e-learning as the use of multimedia technologies in order to develop and improve new learning strategies. It includes the use of informatics tools, such as CD-ROMs, internet or mobile devices, to realize a docent’s work via internet [2].

Making use of e-learning technology, the student has access to interactive and multimedia courses in web format, supported by communication means allowing on-line collaboration with regard to the studied subjects [3].

This technology also enables tutoring support by an expert who is in charge of tracking the student’s learning progress and supporting him in matters of orientation, doubts resolution, motivation, etc.

Within the e-learning field, the creation of contents can be regarded as one of the most important activities; it could be stated that the value of e-learning is mainly determined by them. At the moment, there is a tendency to develop small and reusable didactic modules or units that can be combined with each other to be used in different formative plans and in different frameworks.

Recently there is a great problem concerning content development because the greater part follows the known pattern of a book or a face-to-face-class, for example by translating a book’s text to HTML language while totally lacking dynamism and captivation [4].

In other cases, educational materials make excessive use of hypertext, thus causing great confusion and a total loss of orientation among the students. It can be said that the use of new technologies is driven by an antique mentality, because a theoretical learning model is actually preferred to the real and natural learning model that can be observed among persons: practical learning and learning by doing.

2. The importance of using simulations in learning
Some research has been done on efficiency of simulation-based learning [5-7], and results revealed an enormous effect on understanding and remembering relevant contents for the students.

Simulations attract the students’ attention while encouraging them to memorize and apply the things they have learned; besides, they enable people with a broad range of learning styles to reach individual learning objectives according to their own rhythm.

Computer simulations are programs that contain a model of a real system [6]. The student can explore
main characteristics of the system by changing values of input variables and observing resulting changes in values of output variables.

At first, input and output of the simulation environments were quite limited, but now increasingly sophisticated interfaces with direct manipulation as input, as well as graphs and animations as outputs are emerging as latest developments in virtual reality environments.

Nowadays there actually exist theories, models and technologies to create learning content that simulates real life working situations, where students must reach certain objectives, resolve difficult situations and show an active behaviour.

In contrast to real life, they do not have to be afraid of making mistakes, which is due to the fact that the system is just an experimental field to construct and explore decision alternatives.

There is a vast amount of simulation’s classifications, but a determinant classification is not instituted yet. Some are based on the learning’s type [8, 9], others on concrete values of simulation variables, time, behaviour, etc. [10], in the same way is possible to find to many others classifications.

For the purposes of our research, we will consider simulations showing two principal characteristics: 1) the student can interact with the simulation, and 2) the simulation is connected to an Intelligent Tutoring System (ITS) used to provide guidelines to the student while interacting with the simulation.

3. SCORM

In the context of e-learning standards, there are many organizations promoting specifications and standards for learning-related technologies: ARIADNE, IEEE, AICC and EDUCASE IMS Consortium [11] can be considered the most important ones.

The U.S. Department of Defence and its Advanced Distributed Learning (ADL) Initiative have integrated the work of all those specifications and standards to form a common and usable “Reference Model”, known as “Sharable Content Object Reference Model” (SCORM) [12].

SCORM is a fundamental reference model upon which everyone can develop models of learning content and delivery. Based on the specifications and standards of the above-mentioned groups, SCORM provides a framework and a detailed implementation reference to enable content, technology and systems to “talk” to each other [11].

3.1. SCORM Sequencing

One primary topic of this investigation about SCORM is the kind of sequencing that it provides to Sharable Content Objects (SCOs).

SCORM employs a rule-based sequencing model [13], where each sequencing rule consists of a set of conditions and a corresponding action. The conditions are evaluated using tracking information (Tracking Model) associated with the activity. The behaviour associated with the rule’s action is performed if the rule’s condition set evaluates to True.

Figures 1 and 2 depict the structure (if [condition_set] then [action]) of a Sequencing Rule.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attempted</td>
<td>Objective Measure</td>
</tr>
<tr>
<td>Objective Status</td>
<td>Objective Measure</td>
</tr>
<tr>
<td>Know</td>
<td>Greater Than</td>
</tr>
<tr>
<td>Satisfied</td>
<td>Objective Measure</td>
</tr>
<tr>
<td>Objective Progress</td>
<td>Less Than</td>
</tr>
<tr>
<td>Know</td>
<td></td>
</tr>
<tr>
<td>Completed</td>
<td>Attempt Limit</td>
</tr>
<tr>
<td></td>
<td>Exceeded</td>
</tr>
</tbody>
</table>

**Figure 1. Sequencing rules conditions**

<table>
<thead>
<tr>
<th>Precondition Actions</th>
<th>Postcondition Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip</td>
<td>Exit Parent</td>
</tr>
<tr>
<td>Disabled</td>
<td>Exit All</td>
</tr>
<tr>
<td>Hidden From Choice</td>
<td>Retry</td>
</tr>
<tr>
<td>Stop Forward</td>
<td>Retry All</td>
</tr>
<tr>
<td>Traversal</td>
<td></td>
</tr>
<tr>
<td>Exit Actions</td>
<td>Continue</td>
</tr>
<tr>
<td>Exit</td>
<td>Previous</td>
</tr>
</tbody>
</table>

**Figure 2. Sequencing rules actions**

Sequencing rules are used for defining sequencing to the learning activities. For example, if the instructional designer wants to let the student “skip” contents which he already dominates, then he can use “If satisfied, then Skip”.

3.2. Deficiencies of SCORM

Although SCORM is the currently most employed standard for e-learning systems to share learning resources, after an exhaustive research about
SCORM’s components we found some deficiencies that do not allow an appropriate treatment of simulations when including them in e-learning systems. The main deficiencies are the following:

SCORM is based on the IMS Simple Sequencing (SS) Specification [14], which has significant limitations. In particular, it does not provide artificial intelligence-based sequencing, sequencing of embedded simulations, collaborative learning, or synchronization between multiple parallel learning activities [13].

IMS SS recognizes only the role of the student and does not define sequencing capabilities that utilize or depend on other actors, such as instructors or mentors. SCORM does not prohibit the involvement of other actors; however, it does not define neither roles of other actors nor their sequencing behaviours [13]. This can complicate the inclusion and handling of Intelligent Tutoring Systems in simulations that can be very useful in specific simulation contexts.

The tracking Model does not register sufficient student’s information, ignoring important data about learning achievements [13, 15].

4. Current proposals for the integration of simulations with SCORM

Some institutions have developed proposals to integrate simulations with SCORM by making use of different architectures and mechanisms [16, 17], for example HLA [18], which is a standard architecture for connecting computer-based simulation systems.

In a common way, all proposed prototypes employ a SCO that launches a “middleware” package (specific for each architecture), which initializes a data mapping process to send information about the simulation and the student’s performance to a SCORM conformant LMS.

In general, the operational sequence followed by the prototypes looks like that:
1. The LMS launches a SCORM conformant SCO
2. The SCO launches a packet that contains the middleware which initializes the simulation
3. The student interacts with the simulation
4. After terminating, the simulation sends information relative to its state and the student to the SCO, making use of the middleware
5. The SCO evaluates the student’s actions, and sends the information to the LMS by means of SCORM
6. The LMS remedies if necessary or determines the next SCO to launch.

Figure 3 depicts the information flow established in common integration proposals.

Figure 3. Prototypes information flow

It is remarkable that these proposed prototypes actually achieve an interchange of data between simulations and a SCORM conformant LMS, however, the employed mapping process might become very complex and can consume a lot of development time [19].

Besides, with respect to these proposals, the simulations and a SCORM conformant LMS are not really integrated because they actually work separately. That means that the simulation represents a “black box” for the LMS, which has to wait until the simulation starts or finishes its run-time process in order to send or receive information respectively. Therefore, during run-time, the LMS can never influence the internal course of the simulation, thus making impossible any kind of tutoring for the student.

5. Proposal of extension to SCORM

We consider a real integration of simulations with SCORM to be a better alternative, making use of mechanisms that allow to “observe” the student’s actions while interacting with a simulation, then taking tutoring decisions or intervene if necessary (based on the observations), and/or changing the course of the simulation.

In order to reach those objectives, this work proposes an extension to SCORM by means of an architecture and a model for the students sequencing while interacting with a simulation.

The architecture includes a Tutoring Module for Simulations (TMS) in there will be integrated a new set of tutoring sequencing rules specific to each simulation. The student’s actions (while interacting with a simulation) will be evaluated by those rules. The new rules will have two purposes: 1) Provide tutoring to the student and 2) Allow affecting the simulation course based on the student’s actions and the student’s model (see figure 4).
Analogously to the sequencing mechanisms used by a SCORM conformant LMS, the TMS (extern to the simulation) will be responsible for tracking. For this purpose it will be supported by a “Detector” Module, by means of which the TMS will receive advertisements indicating “meaningful events” that occur within the simulation (e.g. incorrect student actions, abnormal states of simulation, etc.).

The TMS will also be supported by the Student Model component, which will capture session’s information referent to: general information of student (background, personal data, etc.), student’s actions and current state of learning objectives with respect to the simulation, etc. Once the session has been finished, this information will be sent to the LMS.

As a “responsible” for applying the outcome actions of rule’s evaluations, an Acting Module will be implemented. For example, it will send tutoring advertisements to the student, block and unblock a component of the simulation, determine the next scenarios to be posed, etc.

It is important to point out that all of the architecture’s components will be integrated together with the simulation within the SCO; therefore, they will be executed at the client’s side.

Moreover, the proposed architecture will have a generic part and a specific part as well. The generic part includes the architecture itself, the specification’s interfaces that will be implemented for communication and interoperation of modules, the defined classification and language for the tutoring sequencing rules, the working mechanisms modules of the modules, etc.

The specific part (components with weft in figure 4) is determined by the individual characteristics of specific simulations that must be taken into consideration in order to create the tutoring sequencing rules, the definition and detection of “meaningful events”, etc.

5.1. Tutoring sequencing rules

In a similar way to the SCORM sequencing rules (see section 3.1), the tutoring sequencing rules are composed of conditions and actions.

Based on a realized analysis of the most important features that could be relevant for sequencing decisions of the tutorial module in the course of the simulation, the conditions have been classified into three groups, while the actions have been divided into six ones.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>- State of simulation</td>
<td>- Relative to student</td>
</tr>
<tr>
<td>- Student’s actions</td>
<td>- Over simulation</td>
</tr>
<tr>
<td>- Timers</td>
<td>- Detector</td>
</tr>
<tr>
<td>- Detector</td>
<td>- Student’s Model</td>
</tr>
<tr>
<td>- Sequencing</td>
<td>- Exit</td>
</tr>
</tbody>
</table>

5.2. Detection of events

The Detector will use observers to monitor and detect events by making use of two basic mechanisms:

- **Active wait (polling):** used to constantly monitor the value of a variable within simulation.
- **Call-backs:** used to capture a simulation’s event.

The kind of monitoring will depend on access level to the code and structure of simulated system. Accordingly, three different levels are defined:

1. **Access to source code.** Observers are incorporated into the simulation and the Detector retransmits the notification to the Tutoring Module.

2. **Access to object code.** Observers are incorporated into the simulation but the Detector does not retransmit the notification to the Tutoring Module.

3. **Access to event code.** Observers are incorporated into the simulation and the Detector retransmits the notification to the Tutoring Module.

4. **Access to network code.** Observers are incorporated into the simulation and the Detector retransmits the notification to the Tutoring Module.
2. **Access to state.** If it is only possible to access the state of the simulation (e.g. using an interface), the observers will be located within the Detector.

![Diagram showing monitoring with access to the state](image)

**Figure 6. Monitoring with access to the state**

3. **Access to structure** (objects and relations). If a deeper control on state and events objects of the simulation is required, the Acting Module will have more complex processes, for example: to modify variables and structure, to build a scenario, configuration, etc.

![Diagram showing monitoring with access to the structure](image)

**Figure 7. Monitoring with access to the structure**

6. **Case Study**

In order to apply this proposal to a case study, we have developed a simulation that has the learning purpose to allow the student to learn some basic traffic rules.

The simulation has three difficulty levels that can be reached by the student depending of his learning progress.

When including the simulation into the proposed architecture, the “meaningful events” to be detected will be for example: when the student offends against a traffic sign; when he drives with excessive velocity; when there exists the possibility of passing a car but the student does not do it (omission error), etc.

By evaluating tutoring sequencing rules, it will be possible to send preventive or reinforcing messages to the student, block some simulation component (accelerator, steering, etc.) in order to prevent an accident, store the student’s errors, etc.

### Tables 2 and 3 contain a set of conditions and actions that could be defined for this case, employing the classification established before (Table 1).

**Table 2. Tutoring sequencing rules conditions**

<table>
<thead>
<tr>
<th>State of simulation</th>
<th>Timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Possibility_pass_car</td>
<td>Timer_passing_car</td>
</tr>
<tr>
<td>Velocity_limited_to_X</td>
<td>End_timer_passing_car</td>
</tr>
<tr>
<td>Next_curve</td>
<td>End timer preevaluation</td>
</tr>
</tbody>
</table>

**Table 3. Tutoring sequencing rules actions**

<table>
<thead>
<tr>
<th>To the Student</th>
<th>About simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send preventive message</td>
<td>Block component</td>
</tr>
<tr>
<td>Send reinforcing message</td>
<td>Modify state variable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student Model</th>
<th>Exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store penalization</td>
<td>Save current state</td>
</tr>
<tr>
<td>Store error</td>
<td>Send data to LMS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Detector</th>
<th>Sequencing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate Timer_passing_car</td>
<td>Preevaluate</td>
</tr>
<tr>
<td>Annul Timer_passing_car</td>
<td>Send to next level</td>
</tr>
</tbody>
</table>

In the following, some tutoring sequencing rules related to the case study are presented. These rules are intended for dealing with specific situations where a student can pass a car:

R1. *if Possibility_pass_car and not Realize_passing and not Waiting_Timer_passing_car then*

   Activate Timer_passing_car, Send preventive message

R2. *if Possibility_pass_car and Realize_passing and Waiting_Timer_passing_car then*

   Annul Timer_passing_car, Store penalization, Send reinforcing message

R3. *if Possibility_pass_car and not Realize_passing and End_Timer_passing_car then*

   Store error, Send reinforcing message

R4. *if not Possibility_pass_car and not Realize_passing and Waiting_Timer_passing_car then*

   Annul Timer_passing_car, Store penalization

7. **Conclusions and future work**

It is important to point out that this proposal pretends to become compliant with the SCORM philosophy. Therefore, it will assume the terminology and structure conformant to this standard, so that a LMS and the sequencing in simulation can work in an analogous way.
We expect the development of this architecture to lead to many advances in e-learning systems and in general in Educational Informatics, since the inclusion of simulations as learning objects in teaching platforms offers a more effective and stimulant learning to the students. Moreover, a LMS based on SCORM will have a new mechanism making use of the MTS that allows intervening during run-time simulations in order to monitor the student’s behaviour, to provide him sequencing and to influence the course of simulation.

At present, our investigations are still in progress, the major part of the architecture components has been implemented and the whole architecture will be implemented in short term.

8. References


[2] WIKIPEDIA La enciclopedia libre. (es.wikipedia.org)


