@tic
revista d'innovació educativa
VNIVERSITAT DE VALÈNCIA

A new tool to create on-line examinations. Integration into Aula Virtual

Paloma Moreno Clarí
Profesora Ayudante del Departament d'Informàtica de la Universitat de València
Paloma.Moreno@uv.es

Miguel Arevalillo Herráez
Profesor Contratado Doctor del Departament d'Informàtica de la Universitat de València
Miguel.Arevalillo@uv.es

José Ramón Carrillo Cester
Estudiante de Ingeniería Técnica de Telecomunicación Especialidad Telemática en la Universitat de València
jocaces@alumni.uv.es

Vicente Cerverón Lleó
Profesor titular del Departament d'Informàtica y director de la Escola Tènica Superior d'Enginyeria de la Universitat de València
Vicente.Cerveron@uv.es

| **Fecha presentación:** 07/05/2009 | **Aceptación:** 09/06/2009 | **Publicación:** 06/07/2009

Resumen

Actualmente, la Universitat de València utiliza un Sistema de Gestión del Aprendizaje (*Learning Management System*, LMS), como apoyo al proceso de enseñanza-aprendizaje. El LMS incluye una herramienta que permite a los profesores crear exámenes online. Sin embargo, la interfaz de usuario que provee la plataforma no presenta un entorno amigable. Para mejorar la usabilidad del módulo, hemos desarrollado un programa informático que simplifica el proceso. Partiendo de un programa open source, denominado QAED (desarrollado en la Universitat Pompeu Fabra), lo hemos desarrollado para permitir a los profesores crear exámenes offline. Después, estas pruebas podrán importarse fácilmente a la plataforma *Aula Virtual*.

Palabras clave: e-learning, dotLRN, evaluación, examen online

Resum

Actualment, la Universitat de València utilitza un Sistema de Gestió de l'Aprenentatge (*Learning Management System*, LMS), de suport al procés d'ensenyament i aprenentatge. El LMS inclou una eina que permet als professors dissenyar exàmens online. Tanmateix, la interfície d'usuari que proveeix la plataforma no resulta gens amigable. Amb l'objectiu de millorar la usabilitat del mòdul, hem desenvolupat un programa informàtic que simplifica el procés. Partint d'un programa open source denominat QAED, (desenvolupat a la Universitat Pompeu Fabra), l'hem desenvolupat per a permetre als professors crear exàmens offline. Després, aquestes proves podran fàcilment importar-se a la plataforma *Aula Virtual*.

Paraules clau: e-learning, dotLRN, avaluació, examen online

Abstract

Universitat de València actually uses a Learning Management System (LMS) to support teaching and learning activities. This LMS includes a functionality which allows teachers to create online evaluations. Unfortunately, the platform does not provide a user friendly interface. In an attempt to improve the usability of the module, we have developed a computer program which simplifies the process. We have started from an existing open source program, called QAED (developed at Universitat Pompeu Fabra), and developed it to allow teachers to create tests offline. These tests can then be easily imported into the platform *Aula Virtual*.

Keywords: e-learning, dotLRN, assessment, online evaluation

1. Introduction

In 2003, Universitat de València decided to develop Learning Management System, (LMS) to support the teaching and learning process. After examining the existing (free and non-free) products (Roig 2003), Universitat de València decided to join the dotLRN project (Essa *et al.* 2005) and adapted the product to its specific needs (Cerverón *et al.* 2006). The platform resulting was called *Aula Virtual*, and is currently used to provide several functionalities for every single course given in the University. These services may include, for example, a documentary repository, an agenda, a news section, forums, chats, e-mail, services, notification services, work submission, a learning objects repository, an application to create Web presentations, weblogs, a photo album, a FAQs section, etc.) (Cerverón *et al.* 2007).

At the same time, Universitat de València initiated a program devoted to the continuous improvement of *Aula Virtual*. In this program, we evaluated the Assessment module. In particular, we focused on a tool which allows teachers to setup online examinations. This module allows the teacher to select the type of answer (multiple choice, short answer, long answer and uploaded from an external file); fix some presentation related parameters (order, number of questions per page); establish the duration of the test; reuse components (questions, sections and answers); and import and export questions and/or complete tests.

As a result of our analysis we concluded that, although this is an excellent tool for monitoring student progress, its use is still insufficient. One of the main reasons identified was the fact that this tool is not user-friendly at all.. Consequently (and in order to encourage the use of this module), it was decided to simplify the creation of online examinations. And the easiest and most effective solution found was the use of an external tool, which allowed teachers to create online examinations, and export them into *Aula Virtual*. Universitat de València was using *Respondus* application for this purpose (Del Ramo et al. 2008), a non-free software which also allow the user to edit, import, and export examinations created with other applications. The main disadvantage of this product is the cost of the license. That's why it was decided to replace this software by an open source tool, which may provide the same functionality.

As a result of an investigation on existing tools, it was found out that the University Pompeu Fabra had already produced a java open source application called QAED (Martínez 2005). This program allows the user to create multiple choice assessments, and to import and export them from/to version 2 of the ims-qti specification (by using Java library named qti-lite) (Savago, 2005). Although this tool didn't implemented part of the functionalities required by the Universitat de València, it was considered a good candidate for extension, and this was a better alternative than starting a new application from scratch.

2. Analysis

Once the aim of the project had been established, an extensive study was carried out, in which the two tools involved were analyzed (dotLRN's assessment module and QAED).

2.1 Requirements

The analysis of dotLRN's assessment module has produced a list of requirements that the extended version of QAED should offer, in order to fulfil the particular needs of the Universitat de València. A first difference in functionality is that the assessment module offers various types of questions, unlike QAED which only offers multiple choices. Besides, assessments created in dotLRN contain many options which can be activated or modified by teachers, in order to adjust the assessments to the type of evaluation process that best suits their needs.

Another important difference is in the export function. dotLRN generates an XML file which complies with version 1 of the ims-qti specification (compressed in ZIP format). Therefore, our extension of QAED must also allow the user to export the assessments in this format (QAED supported version 2 instead).

The last issue to consider is that dotLRN's assessments contain three major components. The basic unit of an assessment is the question. Questions are grouped into sections and an assessment is composed of multiple sections. A question cannot be found inside an assessment unless it belongs to a section. Each section contains a series of parameters which indicate how they should be presented, their score and the time needed for their completion. In QAED, assessments are composed of multiple choice questions only. Then, it's necessary to adapt our extension of QAED, so that assessments will have the same structure as in dotLRN.

2.2 QAED

Once the requirements have been determined, a first stage is to analyze QAED, to fully understand its internal structure and evaluate the effort required to implement the extensions.

QAED was developed in Java. The fact that QAED is a desktop application implies that it uses javax.Swing to implement its graphic interface. The GUI (Graphical User Interface) appears divided into three sections. The menu bar at the top can be used to edit the assessments. Below this bar, there are two panels. The left panel contains a tree that visualizes the structure of the assessments that are currently open. The right panel shows the possible actions and content of the components that are selected in the tree.

The original version of QAED (QAED Link) uses three libraries, namely JDOM, Xerces, Xalan and jQTI-Lite. These libraries are used to manage XML documents and will be the ones used to provide import and export functionalities. JDOM, Xerces and Xalan facilitate the generation of XML documents (to export files) and the extraction of information from existing ones (to import files). JQTI-Lite is a library which is used to import and export documents in a format compatible with version 2 of the ims-qti specification.

3. Design

3.1. Data Structure

A first design decision concerns the data structure required to support the new components that dotLRN offers. The most logical and intuitive structure is to consider a Java class per type of component: *Assessment*, *Section and Question*. Because questions can be of different types, the class Question is declared abstract and a subclass is created for each type of question supported: *MultipleChoiceQuestion, ShortAnswerQuestion, LongAnswerQuestion, UploadedAnswerQuestion* (for questions which require the upload of an external file). This class structure (illustrated in Figure 1) is used to store information concerning the different components. This information is obtained as user input (using data forms) or from XML files (by importing data) and used to execute the assessments or to export them in other format.

Collections (represented by a black diamond in the diagram above) have been implemented using the class java.util.List. The root component for the structure is the class *Repository*. This class was already implemented in the previous version of QAED, but has been modified to support the extensions. A repository may contain folders (a recursive

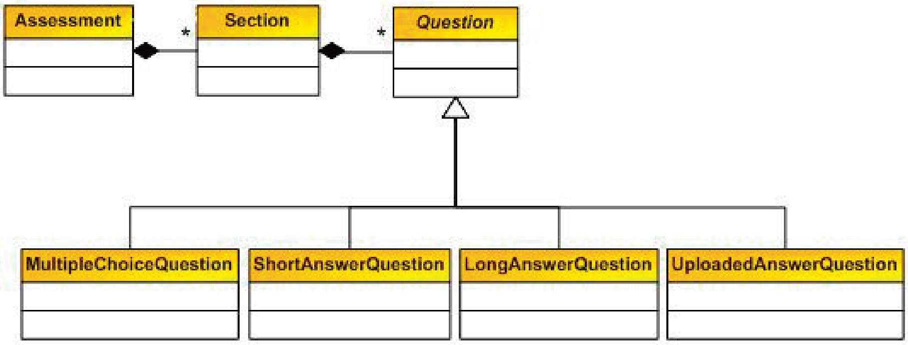

Figure 1. UML static class diagram that represents some of the major applications classes required to store an assessment.

structure which may also contain other folders, apart from assessments and sections). Both *Assessment* and *Section* objects can be directly linked to a *Repository* or to a *Folder*. A repository can also be stored in a file on a PC.

3.2. Graphical User Interface

3.2.1. Menu Bar

The next step was to decide how the new functionalities should be incorporated into the existing application. For this purpose, we added a new menu group into the menu bar (called *UV*). This menu contains three second level menus UV Assessment, UV Section and *UV Question*, each providing a set of actions which are active only when a component of that type is selected (except for the case of creating a component, which is always active, as long as a repository has already been created).

New items have also been incorporated into the toolbar which appears just below the menu. These allow us to create new assessments, sections and questions by pressing the button which contains the appropriate icon. Besides, the button which contains an 'x' symbol can be used to eliminate the component that is selected at the time. Additionally, the two buttons with arrows can be used to export and import assessments (the export button will only be available when a UV assessment is selected).

Both the new menu and the buttons use event listeners to carry out the actions. These methods can be found in the main class of the GUI (QTILiteAppMainFrame), along with all the code needed to create the main application window.

3.2.2. Forms and Wizards

Once the menu had been created, the next step was to create the forms that would allow us to create and edit the Universitat de València 's components. To create these components, we decided to use simple wizards (a series of JDialogs that will successively appear one after the other, until all the necessary information has been filled in). In all cases, the first panel contains a tree representing the structure of the entire repository and the user is asked to select the new component's location. Since components can only be placed at certain places depending on their type (questions inside sections, sections inside assessments and assessments inside folders or repositories), error messages are generated if an invalid location is selected.

3.2.2.1. Assessment Creation Wizard

The wizard to construct a new assessment is composed of the following dialogs (which are displayed sequentially): *Title and Description, Instructions, Answer Options, Pages, Timing Options* and *Other Options*. The sequence of forms is explained below.

First, the *Title* and *Description* dialog is displayed. This is the only dialog that contains three buttons (*Cancel, Complete* and *Accept*). The reason for this is that there are two types of assessment, namely *simple* and *expert*. By clicking on the Accept button, a simple assessment is created. Likewise, an expert assessment is created by clicking on the *Complete* button. In both cases the button's listener is triggered and checks that the title field is not empty.

The *Instructions* dialog is opened. This one contains a text area to introduce the information needed by the student to perform the assessment.

The *Answer Options* dialog takes control. This dialog contains three numeric fields and therefore, the Accept button's listener will verify that they contain valid content (an integer).

The *Pages* dialog opens up, providing a set of tabs (JTabbedPane objects) which are browsed to fill in the different pages. In the Start and End Pages there is a rich text editor that will introduce the HTML code it generates into the expected attribute of the corresponding class. To provide this rich text editor, an external open source java library called Ekit has been used.

The *Timing Options* dialog is used to specify the assessment time. Another external java library has been used for this purpose (JCalendar). This is used to insert two calendars inside the dialog, which will allow us to select the assessment's start and end date. Besides, a set of drop down lists is provided to select the start and end time (only of the checkboxes associated with then have been selected). By using a list, the introduction of invalid values is avoided. Once the start and end check boxes are selected, the Accept button listener will verify that the end time and date is after the start time and date.

Finally, the *Other Options* dialog allows the specification of an IP mask. In this case, the *Accept* button listener checks that the IP mask has a valid value.

3.2.2.2. Section Creation Form

To create a new UV section on a selected location, a single dialog is required. This dialog contains a set of tabs (JTabbedPane) which the user needs to browse to specify all different parameters. The main one is the Visualization Mode tab. In QAED, visualization modes determine how sections are presented to students. Each repository has a list of visualization modes, and these can be applied to sections. The Visualization Mode tab provides a drop down list which contains all visualization modes which are available for the current repository, and allow the user to select one of them for the section. This drop down list also contains an option called *New*. If this is selected a new dialog is displayed that allows the user to create a new visualization mode. This dialog is composed of two tabs. The first one contains mandatory fields whereas the second one contains those which are optional.

3.2.2.3. Question Creation Dialog

To create a new question within a section, another simple form is used. Again, a set of tabs (JTabbedPane) allow the user to specify all the parameters required. The form contains two mandatory fields: the description and the type of question (which is selected from a drop down list). Once a question type has been selected from the drop down list, a new panel appears inside the tab. This panel shows the type of question selected and requests the values for the all parameters associated with this type. As an example, if a multiple choice question was selected, it would be necessary to specify if the answer is multiple or unique, the text for each proposed answer and which one/ones is/are correct. In this case, the *Accept* button listener verifies that all mandatory fields have correct values; and stores the values in an appropriate structure for each question type.

3.2.2.4. Edition Dialog

We have also implemented a dialog to edit the components which have been created using the forms and wizard described above. These dialogs are very similar to those used for creation, but present a few exceptions.

First, all dialogs used for editing purposes are internal frames (subclasses of JInternalFrame). This makes it possible to place them inside the panel located at the right hand side of the main frame. Second, the edition of an assessment is not performed using a wizard. Instead, the equivalent functionality has been provided using a set of tabs in a single form. Finally, we have introduced a new tab in the cases of assessments and sections. In the assessments, the new tab includes information on the sections it contains, and allows the user to modify their contents. Likewise, the new tab in the section's dialog includes information on the questions it contains, and allows the user to edit them.

In both cases, the new tab is composed of a drop down list and five buttons, namely *edit, remove, add, move* and *new*. The *edit* button opens an edition dialog for the selected question or section and it allows us to edit all its attributes. The *remove* button eliminates the selected question or section from the list of the components that is being edited.

The *add* button opens up a dialog in which we will find the repository's tree and which will allow us to add a question or section to the component we are editing. The *move* button moves or copies the selected component to a location in the repository tree (erasing it or not from its previous location). Finally, the *new* button opens the creation dialog for the selected question or section.

3.3 Import and Export functions

In this section, we describe the implementation of the classes required to support the import and export functions. These classes use an external java library called JDOM. This library provides a series of java classes which are useful to read, write and manipulate XML files. In particular, it allows the programmer to convert an XML file into java objects and vice versa (create valid XML files from a set of java objects).

To export an assessment, the corresponding option is selected from the menu (or from the toolbar located just below). Then, a typical file browser dialog requesting the location and file name is displayed. Once these are specified, the XML file is generated and then this file is compressed into a ZIP file. The XML file generation process involves creating each element (one by one) using JDOM, and storing them in the correct place in the XML file, according to the version 1 of the ims-qti specification.

In the case of the import operation, the XML document is first extracted from the ZIP file selected. This is then used to generate the tree structure of a QAED assessment, using JDOM. Once the import operation is completed, the Assessment java object created is stored at the root of the repository.

4. Conclusions

The *Assessment* module provided by *Aula Virtual* allows lecturers to setup online examinations (Moreno *et al.* 2006). Despite the potential of this tool, it presents a low usage rate (Moreno *et al.* 2008). This is due to an uncomfortable GUI and tot he need to prepare the examinations on-line. To solve these limitations, we have extended an existing open source product (called QAED) to provide an easy to use, open source desktop application which fulfils the objective of creating, exporting and importing assessments that comply with those in the dotLRN assessment module.

Among others, this newly created software offers the following advantages: a) it allows teachers to prepare examinations offline; b) it simplifies the creation process; c) it reduces considerably the time required to prepare an examination; and d) it can be used at no cost, avoiding expensive licenses required by other alternatives.

With this development, we hope that usage rates for this module are increased, and teachers are encouraged to make use of the tool.

References

Cerverón, Vicente; Moreno, Paloma (2006). «Aula virtual»: an e-learning management plat-form of Universitat de València development based on Open Code and Collaborative Software. In: *M-ICTE, IV Conferencia Internacional sobre Multimedia y Tecnologías de la Información y Comunicación en Educación, Current Developments in Technology-Assisted Education*, Vol. II, Badajoz: FORMATEX, 2006.

Cerverón, Vicente; Moreno, Paloma, Cubero, Sergio, Roig, Darío, Roca, Salvador (2007). Universitat de València's *Aula Virtual*: a Single Integrated LMS for a University. In : *IADIS e-Learning 2007 Conference Proceedings*. Lisbon: *IADIS Digital Library, 2007.*

Del Ramo Romero, José; López-Soler, Neus (2008). Respondus: un programa para la creación y publicación de cuestionarios. *@tic. Revista d'innovació Educativa. Universitat de València,* 1, pp. 79-81.

Essa, Anthony; Cerverón, Vicente; Blessius, Conrad (2005). .LRN: An Enterprise Open-Source Learning Management System. In *Proceedings Educause2005,* Orlando.

QAED LINK: http://gti.upf.edu/leteos/newnavs/qaed.html. Fecha de consulta, 10.02.2009.

Martinez, Juanjo (2005). JQTI-Lite Library: Developer's Guide (1). http://www.tecn.upf.es/~ssayag/leteos/jQTILite/docs/jQTILite10_Devel_Guide.pdf. Accessed, 10.02.2009.

Moreno, Paloma; Cerverón, Vicente (2006). Plataforma tecnológica para potenciar los procesos de enseñanza-aprendizaje: desarrollo en la Universitat de València basado en software libre y colaborativo. In *Proc: SIIE 06. VIII Simposio Internacional de informática aplicada a la enseñanza*. 8th International Symposium on Computers in Education, Vol. II. León: 2006 .

Moreno, Paloma; Cerverón, Vicente; López, Agustín; Roig, Darío (2007). .LRN Consortium: International Collaboration for Developing a Learning Management System. Experience from the Universitat de València. In Proc: iNEER- ICEE2007. International Conference on Engineering Education. [Online]. Coimbra: http://icee2007.dei.uc.pt/proceedings/index.html Accessed, 10.02.2009.

Moreno, Paloma; Cerverón, Vicente; Arevalillo, Miguel (2008). Análisis del uso universitario de plataformas de gestión del aprendizaje y su relación con la innovación educativa y la calidad docente en el marco del EEES. Resultados en la Universitat de València y procedimientos para la generalización. In *Proceedings: Conferencia sobre Software Libre en Educación Superior,* Valencia: 2008.

Roig García, Darío, 2003. Proyecto de Evaluación de Plataformas de Teleformación para su Implantación en el Ámbito Universitario (1). http://www.uv.es/ticape/docs/dario/memdariov8.pdf. Accessed, 10.02.2009.

Sayago, Sergio, 2005. User Manual v1.0 QAED: Assessments and Question editor (1.0). http://www.tecn.upf.es/~ssayag/QAed10_User_Manual.pdf. Accessed, 10.02.2009.

| Cita recomendada de este artículo

Moreno Clarí, Paloma; Arevalillo Herráez, Miguel; Carrillo Céster, José Ramón y Cerverón Lleó, Vicente (2009). A new tool to create on-line examinations. Integration into Aula Virtual. *@tic. revista d'innovació educativa.* (nº 2) [Docentic] http://ojs.uv.es/index.php/attic/article/view/78/102 Fecha de consulta, dd/mm/aa