Building SCORM embedded WebLabs with LMS interaction

Ildefonso Ruano Ruano, Javier García Gámez, Juan Gómez Ortega
Group of Robotic Automation and Computer Vision (GRAV)
University of Jaén
Jaén, Spain
{alonso, jggarcia, juango}@ujaen.es

***Abstract*—This paper outlines a new procedure for building educational Web Laboratories (WebLabs). These WebLabs are based on JAVA and are offered to learners from Learning Management Systems (LMSs). The Virtual-Remote Laboratories (VRL) are JAVA Applets embedded in Sharable Content Object Reference Model (SCORM) packages that are able to interact with the LMS where they are hosted. This interaction is based on the exchange of information defined by the Run-Time Environment (RTE) and the Run-Time Navigation (RTN) SCORM data models. Both models are managed by the LMS where the SCORM content has been loaded. There are three types of data: read-only, write-only and read/write. In addition, a new JAVA package called scormRTE is presented; it has been developed to facilitate the VRL-LMS information exchange. This package enables a JAVA applet to manipulate directly all the elements of RTN and RTE data models. For the validation, a VRL, created with Easy Java Simulations (EJS), has been developed using this proposal. This VRL has been embedded into a SCORM package in order to build a virtual WebLab about dynamic system modeling. In particular, it is the modelling and identification of a car's suspension dynamics. This WebLab is used in an Industrial Automation course that is offered in several Engineering Grades of the University of Jaén. It is available in the institutional LMS.**

Keywords— Automatic Control Education; e-learning; WebLabs; Easy Java Simulations; SCORM; Online contents; LMS;

I. Introduction

Laboratories are an essential resource in the degree programs of higher education; particularly in engineering and applied sciences. It can be presented to students in three ways: hands-on labs, remote labs (RL), virtual labs (VL) or mixed labs [1][2]. In the three last cases the use of the Web as user-interface is very common [3]. Moreover, the Learning Management Systems (LMSs) are the main tools used by universities to deliver online courses and augment on-campus courses [4]. They provide many functionalities and tools for the university teaching, but where the LMSs have gained an essential importance is mainly as educational content repositories. In this context, The Virtual/Remote Labs (VRLs) have been inserted in the LMSs as learning resources to take full advantage of the benefits that provide [5][6].

Sharable Content Object Reference Model (SCORM) is the set of standards for e-learning more widely used by the LMSs [7][8]. SCORM contents are reusable, durable, interoperable and accessible. It defines communications between client side content and a host system, which is commonly supported by a LMS. On the basis of SCORM, it is possible that a same content may be used in different LMS and in all cases the content can communicate with the LMS where it is hosted [9]. Therefore, some VRLs have been implemented as embedded SCORM content [10], but except some individual cases as in [11], they do not develop the possibilities of VRL-LMS interaction that provides SCORM. This proposal advances in this sense, presenting a methodology for creating WebLabs.

VRLs included in WebLabs can be developed by using different technologies. However, if the VRL is built with JAVA language, a new JAVA package, called scormRTE, that authors have created to facilitate their interaction ability, can be used. The scormRTE package includes two JAVA classes, which in turn have variables and methods with which one can easily set the exchange of information between VRL and LMS. For the validation, we have created an example of SCORM WebLab which includes a VL created with Easy Java Simulations (EJS)[12]. This example uses the scormRTE package to exchange information with the LMS.

The rest of the paper is organized as follows. In section II, the scormRTE package is presented as a means of access to the SCORM data model; in section III, the proposed procedure for building JAVA WebLabs based on SCORM is explained; in section IV, an example of WebLab is presented. Finally, in Section V, the conclusions and future lines are given.

II. SCORM Data Models And scormRTE Package

There are several SCORM versions: SCORM1.2 is currently the most widespread but the most powerful and modern is SCORM2004. In this paper we have mainly considered SCORM2004, although we have taken into account both versions. In any case, SCORM is composed of three sub-specifications:

A. *Content Aggregation Model (CAM) section*

It specifies how content should be packaged and described.

B. Sequencing and Navigation (SN) section

This section specifies how the learner can navigate between parts of the course. It does not exist for SCORM1.2.

C. Run-Time Environment (RTE) section

This section specifies how content should be launched and how it communicates with the LMS.

There are two types of learning resources that a LMS can launch: assets and SCOs. An asset is a digital media (as text or image) that cannot communicate with the LMS, and a SCO is a collection of one or more assets that should communicate with the LMS using the Run-Time Environment. The SCORM compliant LMSs provide an Application Program Interface (API) that enables the SCO-LMS communication. The RTE API consist on 8 Javascript methods classified into three categories: a) Session methods (Initialize and Terminate) are used to mark the beginning and the end of a communication session between a SCO and an LMS, b) Data-transfer methods (GetValue, SetValue and Commit), are used to exchange data model element values between an SCO and an LMS and c) Support methods (GetDiagnostic, GetLastError and GetErrorString) are used for auxiliary communications (e.g., error handling) between a SCO and an LMS. The RTE data model defines 24 elements, 6 of them collect sets of data related to their respective requirements (they are equivalent to 44 simple variables and 30 array variables). The SN data model defines only 6 elements (6 simple variables).

The new scormRTE JAVA package developed in this work hides the use of Javascript to JAVA programmers. The scormRTE package has been a result of previous work based on the development of JAVA-SCORM integration tools [13]. We have used the netscape.javascript.* package to access the Document Object Model (DOM) of the Web page where the VRL applet is embedded. Thus we can invoke JavaScript methods of the SCORM API as recommended by Oracle (the company responsible for Java) [13]. The scormRTE package has been developed taken into account SCORM 2004 and 1.2 versions, so a VRL can use both versions with minimal changes. That is why scormRTE package has 2 classes, here are provided some numbers that help to understand the dimensions of the created package:

- scormRTE class (v.2004). It includes 74 public variables (30 of them arrays) that correspond to data models and 122 public methods, all of them accessible from the Java VRL that imports it.
- scormRTE12 class (v.1.2). It includes 49 public variables (17 of them arrays) that correspond to data model and 75 public methods, all of them accessible from the Java VRL that imports it.

Additionally, the two classes include other private elements that are necessary for proper operation. The scormRTE methods are classified into three categories as shown below.

A. Invokation method

It is used by most of the others methods to invoke JavaScript functions from JAVA.

B. Session method

They are methods to establish and release SCORM-LMS connections from the JAVA code of the VRL.

C. Data-transfer methods

Most of these methods are related to VRL-LMS data transfer. These methods enable retrieve (and modify if permitted) the value of the information stored in the RTE (SN) data model that is managed by the LMS. These methods can be grouped into two types:

- Methods that enable retrieve values of SCORM data models to use them in the VRL, e.g. the learner name that LMS manages in its data base, the SCORM API version or the execution mode of the SCO where the VRL is embedded. For each of these queries, a specific method was created in order to facilitate its use by the VRL developers, so they have no obligation to know the details of the model and communications that are made.
- Methods that store and/or create new data in the SCORM data models (when possible), depending on the execution of the VRL, e.g. some data sets that are associated with the comments. New comments can be created from the VRL in order to be stored in the LMS, these comments may include a) data strings written by the learner or automatically generated by the VRL, b) timestamp and c) the location.

D. Support methods

These methods enable checking the SCO-LMS communication state by requesting diagnostic and error codes generated by the data transfers.

E. Helper methods

These methods have been created to facilitate the programmers handle data formats required by SCORM models, e.g. data format related to date y time.

F. Constructors methods

Constructors are JAVA methods used to create an instance of a class, also called object. These constructors facilitate the creation of scormRTE (scormRTE12) class objects.

VRL Java programmers who want to use these methods and variables must create a scormRTE (scormRTE12) class object. The scormRTE object facilitates the handling of the RTE (RTN) data model elements and hides the use of JavaScript and SCORM API operation. SCORM2004 is the successor standard to SCORM1.2, so its data model has remained most of the elements included in the SCORM1.2 data model and has added new elements. If a 2004 version data model element has the same functionality as other 1.2 data model element, then the name of the variable in the two classes, scormRTE and scormRTE12, is the same. Similarly, methods that have the same functionality have been created using the same name, the same number of arguments and the same types. Thanks to this, if one has the version SCORM1.2 VRL (using a scormRTE12 object) one can get a SCORM2004 VRL simply changing the type of object scormRTE12 by scormRTE. The reverse process (converting a SCORM2004 VRL to a SCORM1.2 VRL) is provided that the SCORM2004 VRL has not used features not included in SCORM1.2.; e.g. if the SCORM2004 VRL uses an element that does not exist in the SCORM1.2 data model or simply using an element of the RTN data model.

III. Procedure For Building SCORM WebLabs using JAVA

Before undertaking the construction of a laboratory is necessary to think about its objective and the skills one wants the students to acquire after they work on it. We will have to decide which laboratory modality is more effective: VL, RL, hands-on lab or mixed. It is also desirable that the lab were integrated into a complete model of WebLab [14] and offer it to students through their usual learning environment.

A procedure for building a JAVA VRL, and their integration in an LMS embedded as element of a package, SCORM is presented.

A. *Preliminary steps.* You must have installed on the computer the tools to create and run Java programs: the Java Development Kit (JDK) and Java Runtime Environment (JRE). It is also necessary to obtain and save, in the appropriate JAVA library folder, two packages: netscape.javascript and scormRTE.

B. *The VRL Applet code edition.* Each developer has a preference when programming java code. There are many options: you can use a simple editor (e.g. notepad+), a generic development tool (e.g. Eclipse) or an specific software to create simulations (e.g. Easy Java Simulations, EJS). Once selected one, you must perform the following actions in your code:

- To import the scormRTE package. This package imports the netscape.Javascript package.
- The creation of a scormRTE object.
- The programming using the methods and variables of the scormRTE package.

C. *The Applet creation.* Once you have finished the VRL programming, you need to create the applet .jar file of the VRL. There exist different options depending on the tools used.

D. *Signing the Applet.* This step is not mandatory, although convenient. The Applets have execution permissions more restrictive than a normal application. If it is digitally signed, it avoids some of these restrictions and access to local resources is allowed. This requires a certificate and use a program for create the digital signature. JDK includes programs to obtain a certificate and to sign.

E. *Insert Applet in a SCORM package.* To do this, proceed as follows.

- Getting a SCORM package. The easiest way is to modify an existing SCORM package, this involves less work. But there are also editors that can easily create SCORM package.
- Identify the SCO in which to embed the VRL. All SCORM packages have a manifest.xml file. It describes the structure of the content and sequence rules of elements in SCORM package. The Web page of the SCORM package in which you want to insert the VRL must be SCO type in the manifest.xml file.
- Storing the .jar file. The VRL .jar file must be included in the directory structure of the SCORM package.
- The edition of the VRL Web page. In order to insert the VRL Applet in the Web page you must add an <APPLET> HTML element and fill the necessary attributes. E.g: code, codebase, archive, width, height.

F. *Compressing the SCORM package.* The entire structure of the SCORM, the applet jar file included, must be compressed to get a zipped file.

If you have followed these steps, you have created a SCORM package including a VRL. If you want to offer the WebLab to students, the next step you must take is to add and to configure it into a LMS. These steps depend on the LMS used, but should not be very different from the following:

A. *Accessing to the LMS.* Most LMS are accessible via Web. You should open a browser and enter the address of LMS. On the home screen you must enter the credentials (username and password) that allow access. It is necessary that the LMS support the SCORM version that was used to build the WebLab.

B. *Importing the SCORM package.* Before importing the WebLab, you should stay in the virtual space of the LMS (container resource) where you want offer the SCORM package to learners. You also must check that the permissions of the container resource are correct: the tutor must have write permission to add the SCORM module and learners must have the read permission for access to the container resource where the SCORM module is.

C. *SCORM module Configuration.* Once it has finished uploading the WebLab SCORM package at the desired location of the LMS, the configuration options of the SCORM module must be checked. These options depend largely on the model of LMS. Learners must have the read permission to reach and access to the SCORM module.

Following this procedure, a SCORM package including a VRL, which interacts with the LMS, can be created and offered to learners in a LMS. The interaction depends on the methods used in the VRL programming. The most common methods that should be used by all VRL that satisfy this procedure are, at least, those that store the SCO completion status and the score achieved by the learner.

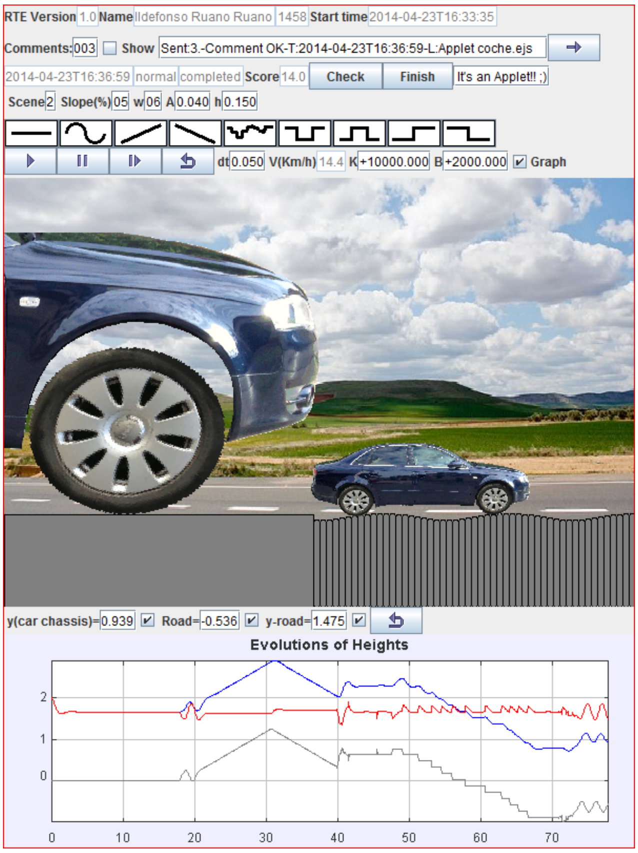
The information that the LMS will show about learner tracking depends on the specific characteristics of the LMS model. Similarly, the access to these data can be done in many ways also depending on the LMS model used.

IV. WebLab Example: Car's Suspension System

We have adapted a SCORM WebLab which includes a VL developed with EJS in order to use this proposal (Fig.1). It is a virtual WebLab on modelling dynamic systems; in particular, it corresponds to the modelling and identification of the dynamics of a car's suspension. This WebLab is used in the "Industrial Automation" course that is included in Engineering Degrees of the University of Jaén and is available in the institutional LMS. The original WebLab performed interaction with the LMS based on a set of specific functions developed ad-hoc [9]. The VRL software has been adapted deleting the specific functions and importing the scormRTE package. Then a scormRTE object (SCORM 2004 version) has been created and several methods have been used for, among others, the following: retrieve from the LMS SCORM mode, RTE version, learner identification, learner name, learner comments, learner score and more. The code of the VRL is now simpler and cleaner and there are more interaction possibilities: now the programmer can manipulate all the data of the SCORM models (RTN and RTE).

V. Conclusions And Future Actions

Within the design of remote laboratories, in this paper we have presented a JAVA package that facilitates the programming of VRL-LMS communications in SCORM environments. This package can be used to work with 1.2 and 2004 SCORM versions. Using this package, the Javascript communication provided by the LMS is hidden to the programmers.



RTE Version 1.0 Name Ildefonso Ruano Ruano 1458 Start time 2014-04-23T16:33:35

Comments: 003 □ Show Sent:3.-Comment OK-T:2014-04-23T16:36:59-L:Applet coche.ejs

2014-04-23T16:36:59 normal completed Score 14.0 Check Finish It's an Applet!! ;)

Scene 2 Slope(%) 05 w 06 A 0.040 h 0.150

dt 0.050 V(Km/h) 14.4 K +10000.000 B +2000.000 ☑ Graph

y(car chassis)= 0.939 ☑ Road= -0.536 ☑ y-road= 1.475 ☑

Fig. 1. Screen capture of the VL

It has been also described a precise methodology that describes how (1) perform programming, (2) to get a VRL Applet, (3) to include an Applet in a SCORM package and (4) to present the SCORM package to learners in a LMS. Using this method, you can obtain packages in the SCORM format capable of interacting with the LMS where it is hosted. This allows, among others actions, the customization of the WebLabs and to perform assessments that can be logged automatically in the LMS.

Acknowledgment

This work was partially supported by the projects DPI2011-27284, TEP2009-5363 and PI10-AGR-6616.

References

[1] Bencomo, S. Dormido. "Control learning: present and future." Annual Reviews in control 28.1 (2004): 115-136.

[2] Ma, Jing, and Jeffrey V. Nickerson. "Hands-on, simulated, and remote laboratories: A comparative literature review." ACM Computing Surveys (CSUR) 38.3 (2006).

[3] Tzafestas, Spyros G. "Teaching control and robotics using the web." Web-Based Control and Robotics Education. Springer Netherlands, 2009. 1-38.

[4] Sun, Pei-Chen, et al. "What drives a successful e-Learning? An empirical investigation of the critical factors influencing learner satisfaction." Computers & Education 50.4 (2008): 1183-1202.

[5] Abdellaoui, N., et al. "Towards the loose coupling between LMS and Remote Laboratories in Online Engineering Education." Education Engineering (EDUCON), 2010 IEEE. IEEE, 2010.

[6] Torre, L. de la, Sánchez, J. P., Heradio, R., Carreras, C., Yuste, M., Sánchez, J. and Dormido, S., "UNEDLABS: An example of EJS labs integration into moodle", World Conference on Physics Education, Istambul, Turkey 2012.

[7] Advanced Distributed Learning, (2012) "SCORM 2004 4th Edition Specification". [Online]. Available: http://www.adlnet.gov/wp-content/uploads/2011/07/SCORM_2004_4ED_v1_1_Doc_Suite.zip

[8] Advanced Distributed Learning (2013). "SCORM adopters". [Online]. Available: http://www.adlnet.org/wp-content/uploads/2012/01/SCORMAdoptersLocked.xlsx

[9] Barra, E., Gordillo, A., Gallego, D., & Quemada, J. "Integration of SCORM packages into web games". Frontiers in Education Conference, 2013 IEEE (pp. 685-690).

[10] Bellver, I., et al. "Packing and reusing virtual web laboratories as Sharable Content Object in wide range of educational fields." Frontiers in Education Conference, 2013 IEEE.

[11] Ruano-Ruano, I., Gomez-Ortega, J., Gamez-Garcia, J., & Estevez-Estevez, E. "Integration of Online Laboratories-LMS via SCORM". Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on (pp. 3163-3167).

[12] F. Esquembre, "Using Easy Java Simulations to create scientific simulations in Java". EUROCON 2003. Computer as a Tool. The IEEE Region 8 Vol:1, 2003, pp.:20-23

[13] Oracle (2013). "Invoking JavaScript code from an applet". [Online]. Available: http://docs.oracle.com/javase/tutorial/deployment/applet/invokingJavaScriptFromApplet.html

[14] Ruano-Ruano, I., Gamez-Garcia, J, Gomez-Ortega, J., "A SCORM Based Package Model for WebLabs". Frontiers in Education Conference, 2014 IEEE. In press.