Interfacing repositories of Learning Objects with support for programming problems

José Paulo Leal [1] and Ricardo Queirós [2]

[1] CRACS & DCC-FCUP, University of Porto, Portugal
zp@dcc.fc.up.pt
[2] CRACS & DI-ESEIG/IPP, Porto, Portugal
ricardo.queiros@eu.ipp.pt

In recent years user expectations regarding repositories of learning objects changed considerably and are increasing. Users are now concerned with issues not completely addressed by the existing systems, such as: compliance with eLearning standards, interoperability with other systems and management of repositories.

This paper presents the interfaces of crimsonHex - the Learning Object (LO) repository of EduJudge. EduJudge is a pan-European project aiming to integrate the collection of problems of the UVA on-line judge in the educational environment. In this paper we focus on recent developments and ongoing work both of the user interface and the application interface of crimsonHex.

The application interface of crimsonHex is a particularly important feature of the repository since it was designed to integrate with other systems in the EduJudge network, namely the evaluation server and the learning management system. For sake of standard compliance these features are based on IMS Digital Repositories Interoperability (DRI) specification. Our experience in using these recommendations lead us to propose extensions to its set of functions and to the XML binding that currently lacks a formal definition.

To evaluate the proposed extensions to the IMS DRI specification and its implementation in crimsonHex, we developed a plugin for the 2.0 release of the popular Moodle LMS. Moodle 2.0 users will be able to download LOs from crimsonHex repositories since this LMS is expected to include the plugin described in this paper in its distribution.

The user interface is also an important part of crimsonHex since it supports all the workflow of an LO lifecycle, from its authoring, trough its validation process, and management in collections, to its selection by teachers. In the design process we identified several usage task profiles - the archivist, the content author, the reviewer and the consumer - and two basic usage profiles.

The user interface was developed using an Ajax framework to enable the implementation of the single screen design. We selected the Google Web Toolkit (GWT), an open source Java software development framework that allows a rapid development of AJAX applications in Java. When the application is deployed, the GWT cross-compiler translates Java classes of the GUI to JavaScript files and guarantees cross-browser portability. The framework supports also asynchronous remote procedure calls. This way, tasks that require high computational resources (e.g., complex searching within the repository) can be triggered asynchronously, increasing the user interface's responsiveness. The complex controls required by the selection and action areas are provided by SmartGWT, a GWT API's for SmartClient, a Rich Internet Application (RIA) system.

We are currently working on the design and implementation of an authoring tool for crimsonHex. Most authoring tools currently available are unnecessary complex and drive away potential authors. Their user interfaces mirror the complex structure of the metadata standards, covering a vast number of facets, from digital rights to technical details, some of which are seldom applicable. Nevertheless, standards are essential to the interoperability of eLearning systems and we believe that it is the role of user interfaces to make complex concepts easier to grasp to the average user. To achieve this goal we designed the authoring tab as a wizard - a common pattern used in graphical interfaces when an application needs to collect a large number of parameters. Wizards use progressive disclosure to present panels with small sets of parameters, and parameters selected in the first panel influence the subsequent panels. In this wizard the uploaded resources are processed and some meta-data, such as file formats and languages, are automatically generated using heuristics. These values are presented in subsequent panels and may be changed by the author.