

# Feeding the beast: Managing your collections of problems

José Paulo Leal<sup>1</sup> and Ricardo Queirós<sup>2</sup>

<sup>1</sup> CRACS & DCC-FCUP, University of Porto, Portugal

<sup>2</sup> CRACS & DI-ESEIG/IPP, Porto, Portugal

## 1 Overview

Although designed primarily for communication with other services, an interoperable repository must also interact with human users. Authors and reviewers of new Learning Objects (LO), students and teachers that browse the collections and archivists responsible for the content of the repository, all these users have different needs and expectations when they interact with the repository.

Usability was the major concern with the user interface design of the repository of learning objects, and to promote it we followed established user interface design principles [1]. The main feature of the resulting design is the use of a single screen common to all user profiles. This type of design breaks with the traditional structure of web interfaces used by other repositories [2].

This chapter focuses on the crimsonHex users interface layer called **Web Manager**. In the remainder of the chapter we detail its main tasks, namely uploading, reviewing, browsing, searching and downloading.

## 2 Managing crimsonHex

The Web Manager is one of the internal components of the crimsonHex repository architecture. The component is included in the repository distribution package. After the installation of the repository (cover in the chapter entitled *Your personal archival Repository Server*), the Web Manager can be accessed through a web browser by typing the following URL: <http://localhost:8080/crimsonHex> where *localhost* is replaced by actual computer's name if access remotely.

The layout of the user interface is shown in Figure 1 where we can identify three main areas:

- Repository view: section where the user navigates through the repository structure to select collections or LO;
- Collection View: section where the user can visualize the contents of the collection selected in the Repository View and select a specific LO;
- Learning Object View: section where the user executes actions (organised by tabs) on the LO selected in one of the previous views.

Secondary areas in this layout are the header, used for authentication and registration, the footer including information about version, license and terms of service of the repository, and sections for news, statistics and help.

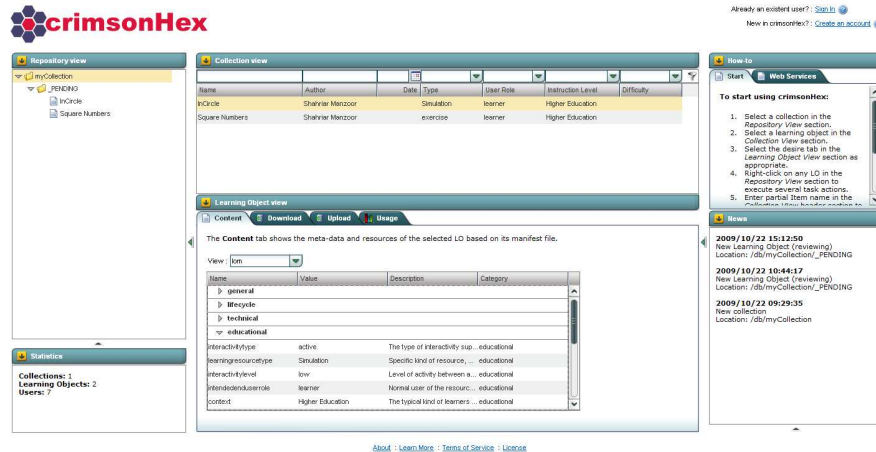


Figure 1: Screen layout of the users interface.

The main feature of this user interface layout is to use a single screen with specific areas for object selection and actions. Object selection is needed by all users, although selectable content depends on the user's profile, thus it can be implemented by a common tree-based control. Different actions require specific forms or panels that also share a common control on the user interface. Since the number of actions is comparatively small we choose a tabbed control to aggregate them. The tab configuration shown to users depends both on their profile and on the current object selection.

As a rule, all available actions have an associated tab, thus helping novice users to recognise which are the available actions. However, some of these actions can be executed directly over selected objects, without requiring additional data. In general, these actions are meant for frequent users and will be bound to contextual menus on the tree-control, as well as to accelerator keys.

The remainder of this section details the actions available in the main areas.

## 2.1 Usage profiles and Authentication

Users of the crimsonHex repository will have different usage profiles. On one hand, many will be novice or first-time users, especially among authors and consumers. On the other hand, we expect some users, especially reviewers and archivists, to use it frequently, tending to become experts in its use. The repository has the following usage profiles:

- **Administrator** - a person responsible for the installation and configuration of the repository;
- **Archivist** - a person responsible for a set of activities related with collection management, such as: creation of collections, assigning of learners and reviewers to collections;

- **Author** - a person that develops and submits LO to the repository. The submission of LO will be enforced to comply with controlled vocabularies defined in meta-data standards (IEEE Learning Object Metadata - LOM) and possible extensions. This class of users will contribute with new learning objects and receive comments from reviewers;
- **Reviewer** - a person that controls the quality of the repository by validating the submitted LO;
- **Consumer** - a person that browses (part) of the repository and has limited access to its content (LO, usage reports, reviews, comments);
- **Guest** – a person not registered in the system and with very limited access to the repository. The profile is automatically assigned when the user enters for the first time in the system.

To assign a usage profile, a user must be authenticated in the system. The authentication is made by clicking in the *Sign In* link located at the top right side of the screen layout and filling the authentication form (Figure 2).

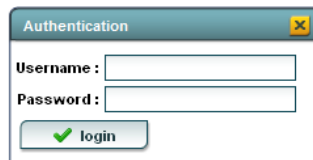

 A screenshot of a web browser window titled "Authentication". It contains two input fields: "Username:" and "Password:". Below these fields is a button with a green checkmark icon and the text "login".

Figure 2: Authentication form

To use the system, a user doesn't need to be registered. By default, when a user enters for the first time in the crimsonHex system a GUEST profile is assigned. The GUEST profile can be configured to have full or limited access to a specific collection in the repository (depends on the Administrator strategy). By default, this collection is called `MyCollection` and is automatically created in a fresh installation of the system.

However, if the user wants to register, he must press the *Create an account* link located at the top right side of the screen layout and filling the registration form as illustrated in Figure 3:

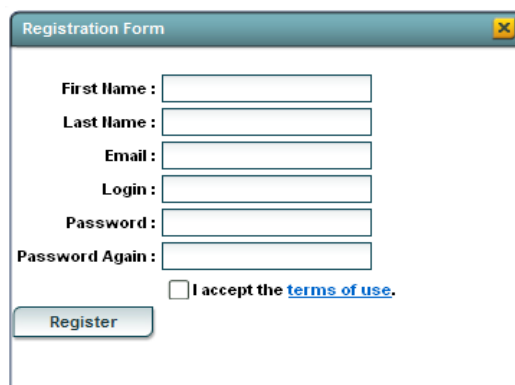
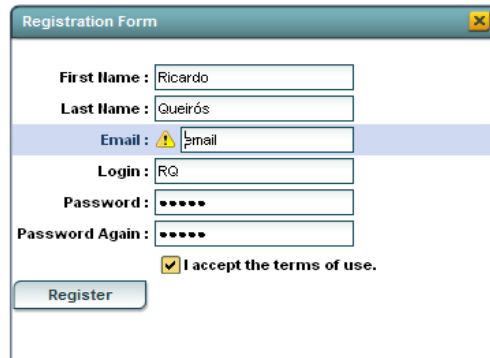

 A screenshot of a web browser window titled "Registration Form". It contains several input fields: "First Name:", "Last Name:", "Email:", "Login:", "Password:", and "Password Again:". Below the "Password Again" field is a checkbox followed by the text "I accept the [terms of use.](#)". At the bottom left of the form is a button labeled "Register".

Figure 3: Registration form

All the fields in the registration form are mandatory and subject to validation. The *terms of use* link refers to a file (by default empty) that must be provided by the Administrator of the repository with the repository terms of use. Figure 4 shows how the Web Manager informs the user of an invalid field:



The screenshot shows a 'Registration Form' window with the following fields and values:

- First Name: Ricardo
- Last Name: Queirós
- Email: [Warning icon] @mail (highlighted in blue)
- Login: RQ
- Password: [Masked with dots]
- Password Again: [Masked with dots]
- I accept the terms of use.
- Register button

Figure 4: Registration form

Pressing the *Register* button, an email is sent to the repository Administrator. After confirmation of this data, the user can fill in the authentication form with the new credentials.



Figure 5: Welcome window

A welcome window (Figure 5) appears at the centre of the screen layout while the user's name appears at the top right side with a *logout* link.

## 2.2 Configuring users

By default, the Web Manager component includes a user with the ADMIN profile and the following credentials: username: `Admin` and password: `changeit`. You can change this data by accessing to the `config.xml` file in the WEB-INF directory. This file describes a set of users that are associated with profiles. A profile defines a set of operations that a user can perform, more precisely:

- **Create** - create new collections;
- **Remove** - remove existent collections or LO;
- **Review** - validate a LO;
- **Comment** - comment about the acceptance or rejection of an existent LO;
- **Upload** - submit a new LO;
- **Download** - download an existent LO;
- **Usage** - show the LO usage data in table or graphical form;

- **Read** - show the LO manifest contents organised in several views defined in the `types` attribute:
  - LOM - shows IEEE LOM meta-data;
  - Extension – shows extension meta-data (e.g. EduJudge);
  - Resources – shows LO resources.
- **View** - show the structure and the contents (collections and LO) of the repository. The visualization is limited to two permissions defined in the `types` attribute: `public` and `private`. The `public` type shows only validated LO, while the `private` type allows the visualization of LO waiting for reviewing.

The default configuration file is shown next.

```
<?xml version="1.0" encoding="UTF-8"?>
<config ...">
  <repository
    host="http://localhost:8080/crimsonHex/lo/"
    admin-email="ricardo.queiros@gmail.com"/>
  <profiles>
    <profile id="ADMIN">
      <item operation="create"/>
      <item operation="remove"/>
      <item operation="review"/>
      <item operation="comment"/>
      <item operation="upload"/>
      <item operation="download"/>
      <item operation="usage"/>
      <item operation="read" types="lom extension resources"/>
      <item operation="view" types="public private"/>
    </profile>
    <profile id="GUEST">
      <item operation="read" types="lom"/>
      <item operation="view" types="public"/>
    </profile>
    ...
  </profiles>
  <users>
    <user
      email=""
      fname=""
      lname=""
      login="Admin"
      password="changeit"
      profile="ADMIN"
      path="/" />
    <user
      email=""
      fname=""
      lname=""
      login=""
      password=""
      profile="GUEST"
      path="/myCollection"/>
    ...
  </users>
</config>
```

To bind a user to an existent profile you must add a new `user` element and set its `profile` element to the relevant profile `id`. In complement to the personal data of the user, this element has also a `path` attribute that defines a sub-tree of collections where commands can be applied.

To use the system, a user doesn't need to be registered. By default, when a user enters for the first time in the crimsonHex system a GUEST profile is assigned. The GUEST profile can be configured to have full or limited access to a specific collection in the repository (depends on the Administrator strategy). By default, this collection is called `MyCollection` and is automatically created in a fresh installation of the system.

### 2.3 Organize collections

In the *Repository View* area the user navigates through the repository using a tree-based control and selects collection and/or LO. This control type was chosen because it reflects the structure of the repository, where LO are grouped in collections that may also contain sub-collections. After authentication, the system automatically assigns a default root collection based on the user's profile. This control displays one column with the name of the object (collection or LO).

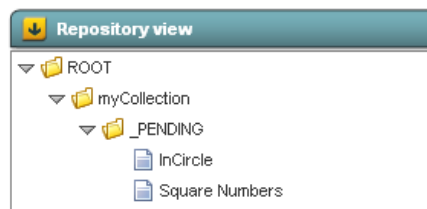


Figure 6: Repository view

After selecting a collection in the *Repository View* the user sees its contents in the *Collection View* section (Figure 7).

The screenshot shows a window titled "Collection view" displaying a table of collection items. The table has the following columns: Name, Author, Date, Type, User Role, Instruction Level, and Difficulty. The data rows are as follows:

Name	Author	Date	Type	User Role	Instruction Level	Difficulty
InCircle	Shahriar Manzoor		Simulation	learner	Higher Education	
Square Numbers	Shahriar Manzoor		exercise	learner	Higher Education	

Figure 7: Collection view

The *Collection View* has a list grid control composed by several columns that can be sorted in ascending and descending order, and new columns can be added to this visualisation as shown in Figure 8.

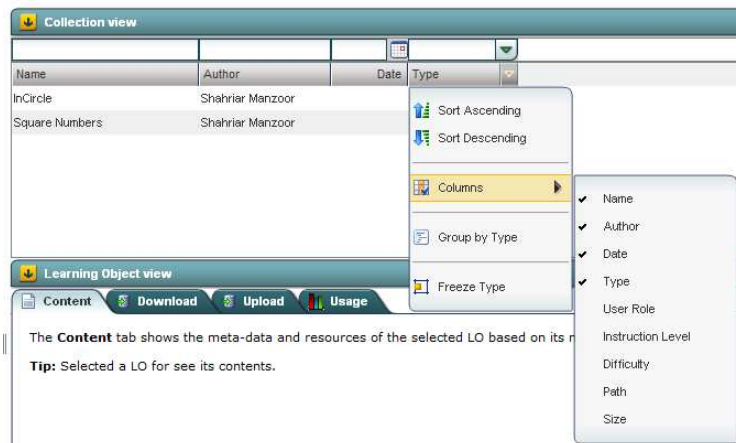


Figure 8: Adding columns to the Collection View

To prepare the repository to receive LO the user starts by creating a collection. To create a collection the user must follow these steps:

1. Select the parent collection of the collection being created;
2. Press the right button of the mouse to trigger a context menu (Figure 9);



Figure 9: Available actions over a selected object

3. Select the *Collection > Create...* option to trigger a new window (Figure 10);
4. Fill the following items in the form:
  - *Collection name*: name of the collection;
  - *Validators*: schemata available for LO validation;
  - *Enable review*: LO review.

Validators:	Name	Class
<input checked="" type="checkbox"/>	EduJudge Metadata Specification v2.0	ConformanceValidatorEdujudge
<input checked="" type="checkbox"/>	IMS Content Packaging v1.2	ConformanceValidatorImscp0

Figure 10: Create collection window

In this version the schemata used for LO validation are the IMS CP schema and the EduJudge Metadata schema. The selection of the *validators* item will influence the submission of LO by enforcing the new LO to comply with the selected controlled vocabularies defined in meta-data standards (IMS CP) and possible extensions (EduJudge).

By checking the *enable review* item, all the LOs submitted to this collection will be stored in a special and contextual collection called *\_PENDING*. The visualization of this collection is only for users who belong to profiles with the following operation element in the configuration file:

```
<item operation="view" types="private"/>
```

To remove collections, you only need to select the specific collection, right click with the mouse and select the *Collection > remove* option. Note that you can only remove empty collections, thus you need to remove its contents first.

## 2.4 Submit a LO

After creating a collection the repository is prepared to receive learning objects. To submit a LO in a specific collection you must:

1. Select the collection (e.g. *Math*) where the new LO will be stored;
2. Select the *Upload* tab in the centre of the screen;
3. Browse for your LO file (e.g. *ChessQueen.zip*);
4. Press the *Upload* button.

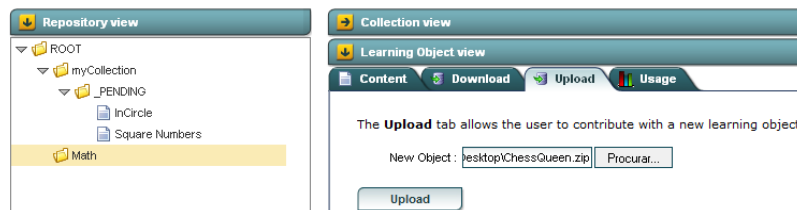


Figure 11: Submission of a LO

A submission of a LO does not imply its immediate availability. As said before, the LO will be subject to a quality assessment by a reviewer. See subsection 2.8 to learn how to validate a pending LO. The following figure shows the LO in its pending stage.

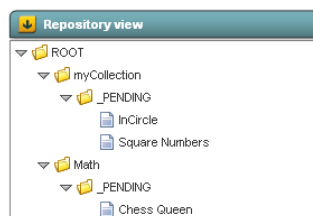


Figure 12: The *\_PENDING* collection



## 2.5 Search a LO

After the selection of a collection in the *Repository View* the user can see its contents in the *Collection View* section (Figure 7). However, in most cases she or he will prefer to search the repository for LO matching certain criteria, either in a specific leaf collection or in top-level collection and all its sub-collections. In this case the user can edit the filter boxes in the *Collection View* header highlighted in a red rectangle as shown in Figure 13. This header allows the user to build a simple query by including field values associated with meta-data elements (LOM and extensions) of the LO manifest.

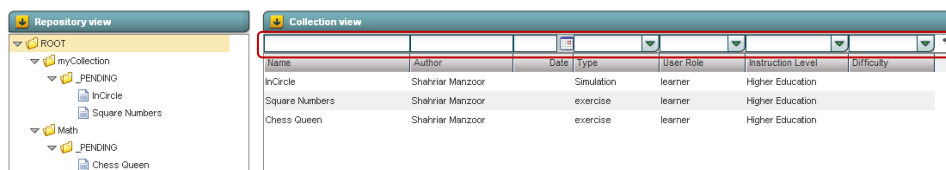


Figure 13: Filter header

This means that the user may start by selecting a collection and then perform a search restricted to that collection. By selecting the root collection the user can search the entire repository. Next, we search for all LOs of the type “exercise” and whose title contains the letters “que”:

1. Select the scope collection (e.g. *Root*);
2. Type the previous values in the specific filter boxes (note: the grid is configured to filter the collection content as you type);
3. Press the filter button at right of the filter header.



Figure 14: Searching LOs

The result of the previous search was one LO (*Chess Queen*). The user can now select it and view its contents in the *Learning Object View* section.

## 2.6 View a LO

The selected LO can be viewed in the *Learning Object View* in different perspectives, such as Content and Usage. As would be expected, each perspective is assigned to a different tab.

The **Content** tab shows the resources and meta-data of the selected LO based on its manifest file. Different views can be selected to configure content presentation, including:

- LOM - shows LOM meta-data grouped in main categories (General, Lifecycle, Technical, Educational, Rights);
- Extension – shows meta-data related with the extension schemata (e.g. EduJudge);

- Resources – lists resources and their meta-data with support for viewing/downloading individual resources.

To view the LOM metadata from the content of the LO the user must follow these steps (Figure 14):

1. Select the specific LO
2. Select the *Content* tab in the *Learning Object View* section
3. Select the LOM option

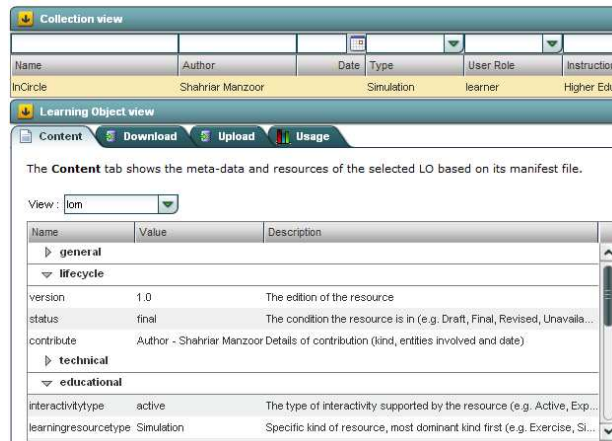


Figure 14: LOM view

The user can also select the extension option to see meta-data related with the extension schemata (e.g. EduJudge) as shown in Figure 15.

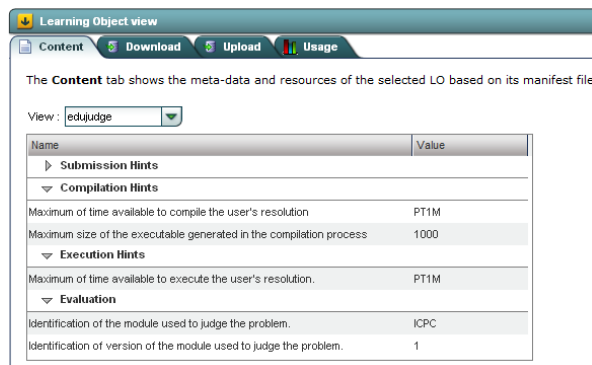


Figure 15: EduJudge view

The user can also select the resources option to list resources and their meta-data with support for viewing/downloading individual resources (Figure 16).

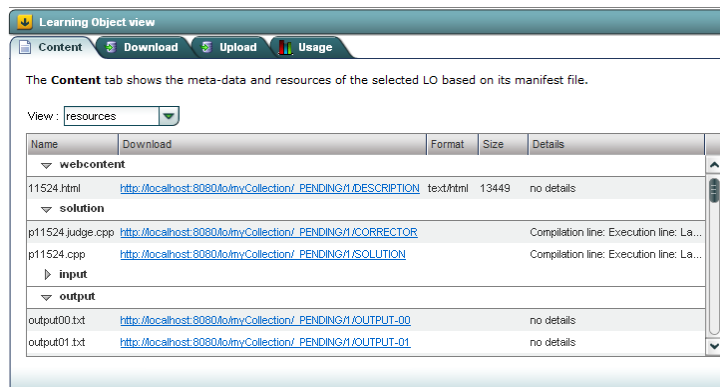


Figure 16: Resources view

In this view you can get a LO asset from the server. An asset could be a problem description, a solution file or even a set of input files from a test case. For download an individual asset just click in the specific asset.

Notice that the Administrator of the repository can control the appearance of these three options by editing the configuration profile. The next lines show the configuration of the Consumer profile inhibiting the LO display only for LOM metadata.

```
<profile id="CONSUMER">
  <item operation="read" types="lom"/>
  <item operation="view" types="public"/>
</profile>
```

Thus, the Administrator can control what parts of a learning object are accessible to the Consumer profile or any other profile.

## 2.7 Download a LO

One of the most important actions in a repository of learning objects is the option to download a LO for personal use or even upload it to a Learning Management System (e.g. Moodle).

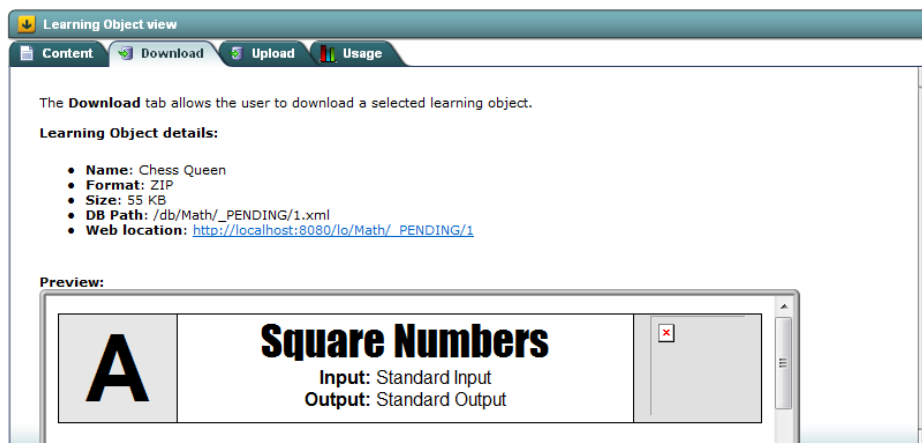


Figure 17: LO download

To download the *Chess Queen* LO follows the next steps:

1. Select the specific LO;
2. Press the *Download* tab (Figure 17);
3. Click on *Web Location* link – a transfer file window appears (Figure 18).

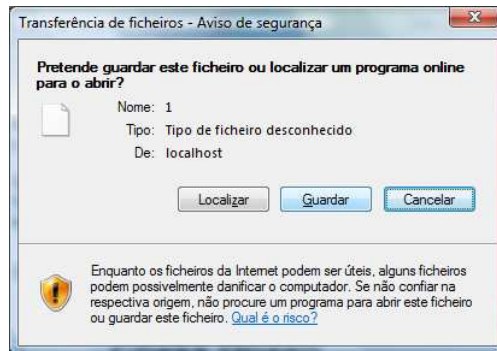


Figure 18: LO download

## 2.8 Analyse usage data

The Usage tab presents statistics on the use of a selected LO. This feature is related to the report function that associates usage reports to an existing LO. This function is invoked by a consumer of the repository services (typically an LMS), summarising an episode of using a LO with a particular student. The aim of this function is to provide the LMS with the ability to dynamically generate presentation orders based on previous uses of LO, instead of using fixed presentation orders. This report includes both general data on the student's attempt to solve the programming exercise (e.g. data, number of evaluations, success) and particular data on the student's characteristics (e.g. gender, age, instructional level). Figure 19 shows the usage data of the selected LO.

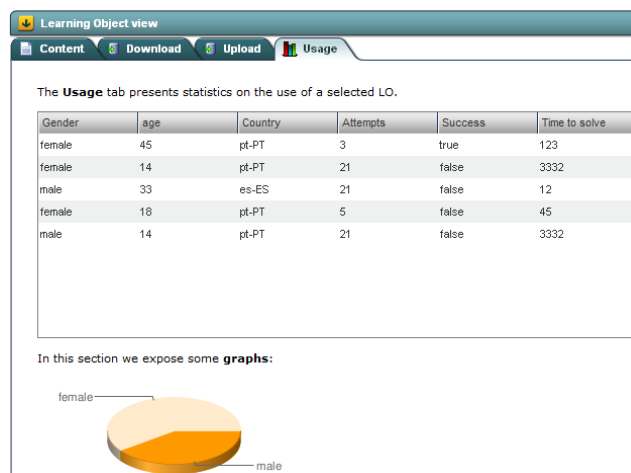


Figure 19: Usage Data

## 2.9 Review a LO

Before validation the LO is not available to general users of the repository. The availability of the LO depends on this validation. If the LO is not accepted then the reviewer should justify the rejection and/or supply comments to the author of the LO by email, for instance<sup>1</sup>. These comments may lead to new versions that must be submitted as a new LO.

To validate the *Chess Queen* LO follow these steps:

1. Select the specific LO;
2. Press the right button of the mouse to trigger a context menu (Figure 20);
3. Select the *LO > Validate* option;

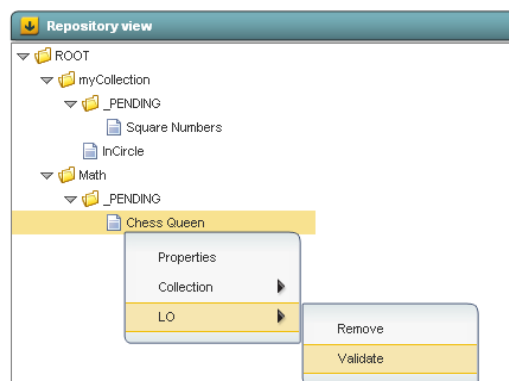


Figure 20: Validation action

After the validation the LO will be stored as child of the *Math* collection and available to registered users of the repository. The Figure 21 shows the visual state of the LO in the repository.

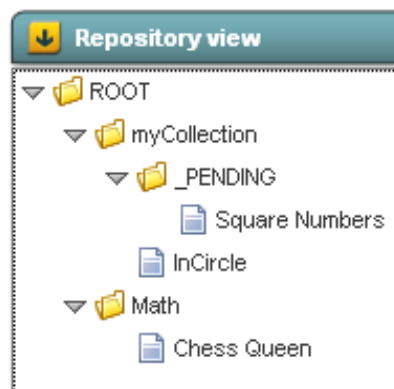


Figure 21: A LO after validation

---

<sup>1</sup> The current version of crimsonHex is not prepared for managing the workflow of LO revision

## 2.10 News and stats

Secondary areas in the Web Manager layout are located in the header (used for authentication and registration purposes), and at the both sides: statistics (left side) and how-to and news (right side).

The **News** Section (Figure 22) displays the recent changes in the state of the repository using an RSS feed. With this option a user can have up-to-date information in the Web Manager or through a feed reader.

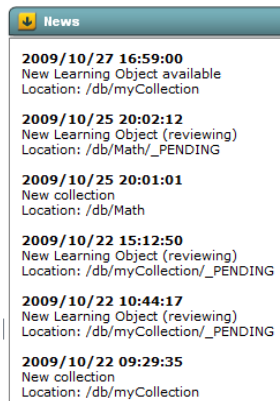


Figure 22: News

The **Statistics** section (Figure 23) provides additional information about the overall content of the repository, more precisely, the number of collections and LO and the number of registered users.



Figure 23: Statistics

The **How-to** section provides basic help on how to use the system and is organised in two tab controls. The first tab (Figure 24) lists the first steps to start using the repository.

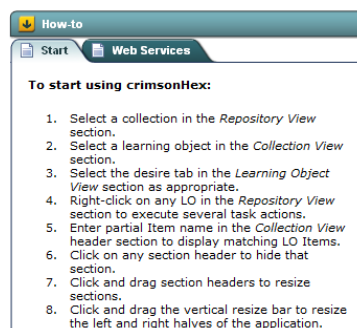


Figure 24: How-to (Start)

The second tab (Figure 25) explains how the user can access the repository through Web Services.

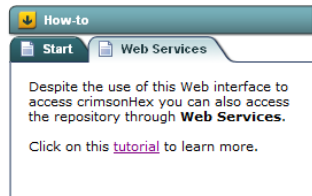


Figure 25: How-to (Web Services)

Interoperability is a main feature of crimsonHex and this user interface provides help on and how to use the core functions of the repository, based on the IMS DRI specification. These functions can be invoked using two web services interfaces: SOAP and REST.

By clicking in the *tutorial* link the user can browse a tutorial (Figure 26) that presents more documentation on accessing the repository using web services.

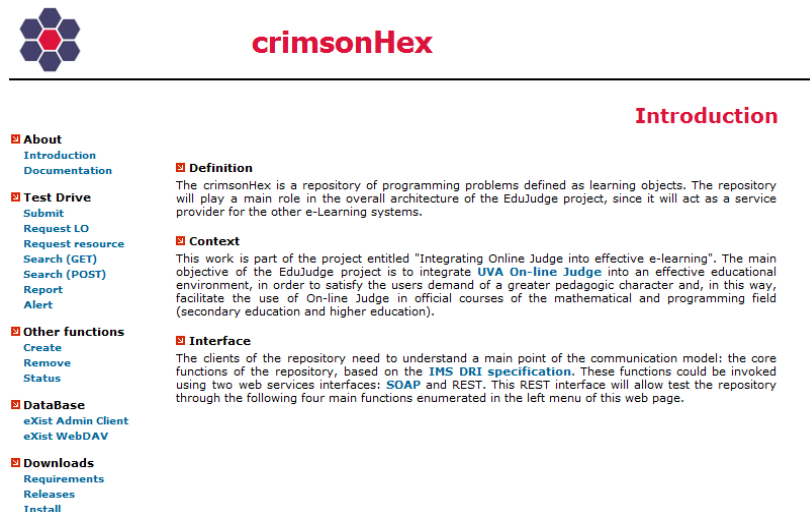


Figure 26: How-to (Web Services)

The tutorial is composed by six sections organised as follows:

- **About** – a brief introduction of the repository;
- **Test Drive** – a list of the main functions (IMS DRI specification) supported by the repository core. This option allows the user to test a specific function using the REST web service;
- **Other functions** – a list of auxiliary functions;
- **DataBase** – tools for accessing the repository database such as the eXist Admin Client and WebDAV protocol;
- **Downloads** – information about the requirements and the steps for installing the repository. This section also lists all the releases of the repository to download (Figure 27) as WAR files;
- **Feedback** – Contacts for comments, suggestions or bug reports.

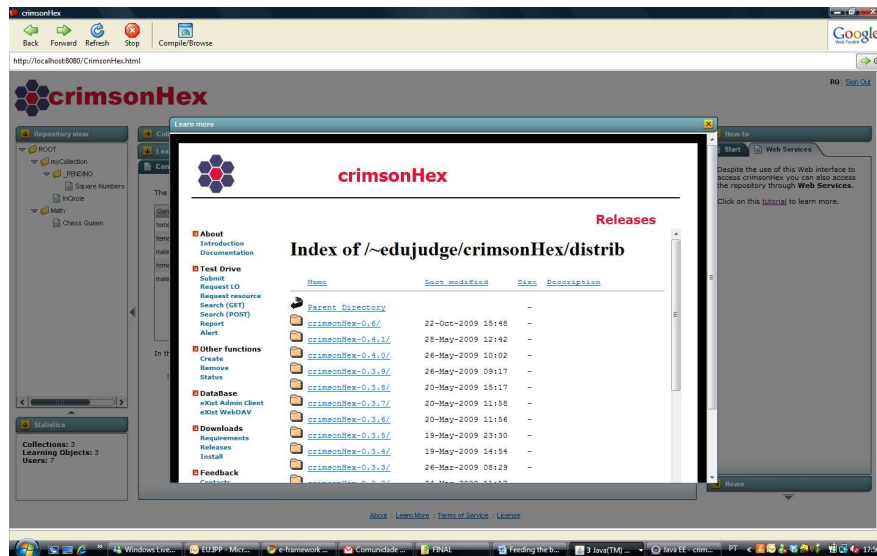


Figure 27: crimsonHex releases

## References

- [1] Shneiderman, B.: Designing the Users Interface - Strategies for Effective Human-Computer Interaction, third edition, Addison-Wesley, ISBN: 0-201-69497-2, (1998)
- [2] Academic ADL Co-Lab: Learning Repositories included in Learning Repository Investigation, <http://www.academiccolab.org/resources/DraftRepositoriesList.pdf>