COLDEX

Collaborative Learning and Distributed Experimentation

Information Society Technologies Programme

Project number: IST-2001-32327

Coldex Portal Access Guidelines (prototype 1)

Deliverable Number:	D.6.2.1
Contractual Date of Delivery:	M18
Delivery Date:	May, 2004
Version:	1.0 (Final)
Lead Partner:	UNED
Contributing partners:	UNED
Authors:	Verdejo, M.F.; Barros, B; Mayorga, J.I.
Contact:	nmayorga@lsi.uned.es

Revision history

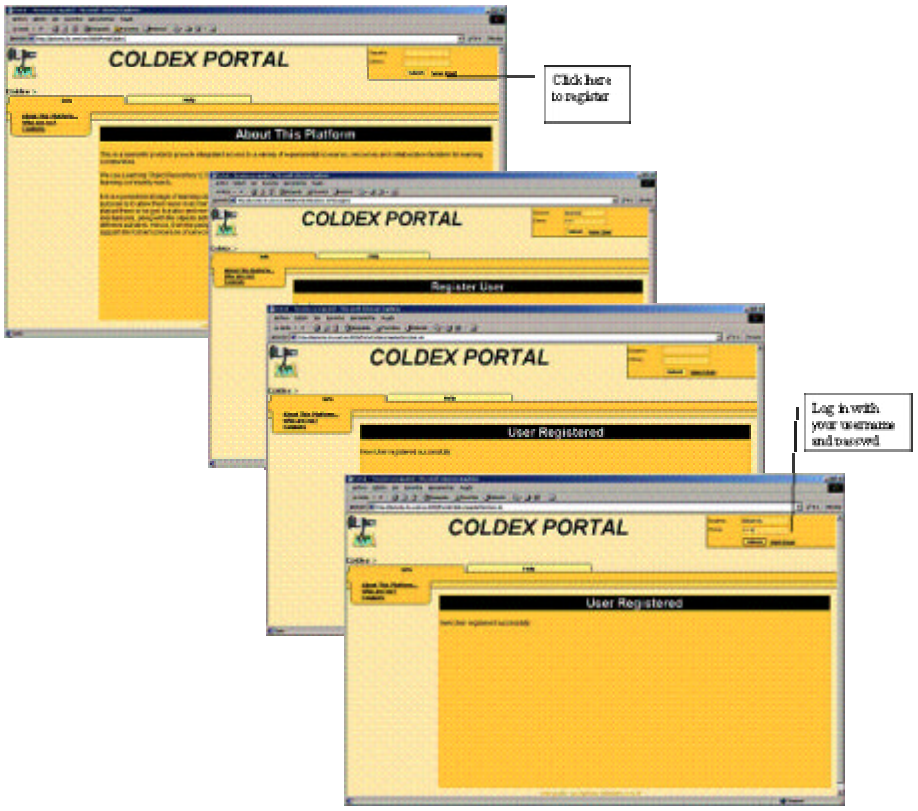
Date	Name	Version	Description	Authors
September 2003	Prototype 0	0	No Distribution First version of the Ontology in KAON	UNED Group
May 2004	Prototype 1	1.0	Second version of the Ontology in Protegé Distribution with LDAP	UNED Group

1 Introduction

Document corresponding to Prototype 1 and which was used, along with the accompanying software in the Open User Scheme in Argentina, in May 2004

2 Register and Entrance

Users must be registered with the portal before being allowed to enter it. A register facility is available for new users. This process is shown in `figure 1` below:

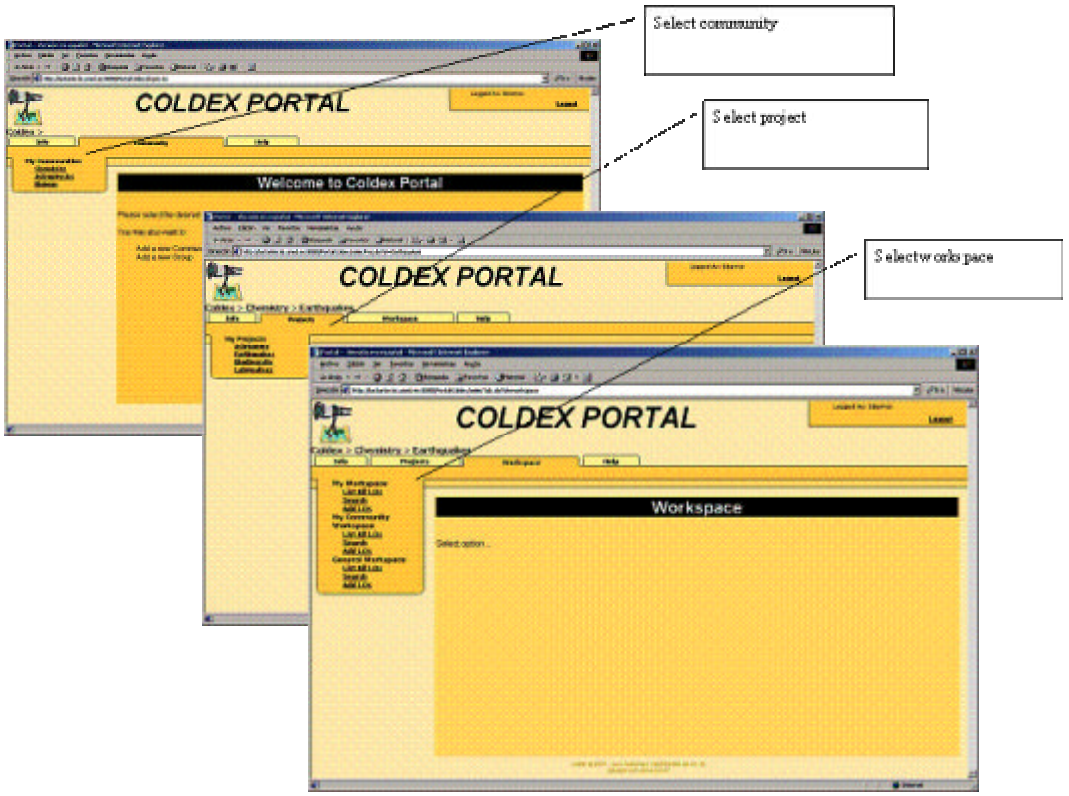


`figure 1: register and login`

This prototype only provides user login and workspace creation. Full management and user activity control will be supplied in the next prototype version.

3 Community, Project and Workspace Selection

Once registered and logged to the system, a user has to choose a **community**,a **project** and a **workspace**. These selectinons can be seen in `figure 2` below:



`figure 2:` chosing community, project and workspace

4 Workspace functionality

There are three types of workspaces: general, for the community and individual. For each workspace, there are, mainly, three available functions: listing all the available Learning Objects, searching Learning Objects and adding a Learning Object. All these operations canbe accessed from the menu./

4.1 Listing Learning Objects

`Figure 3` shows how to list all the available learning objects.

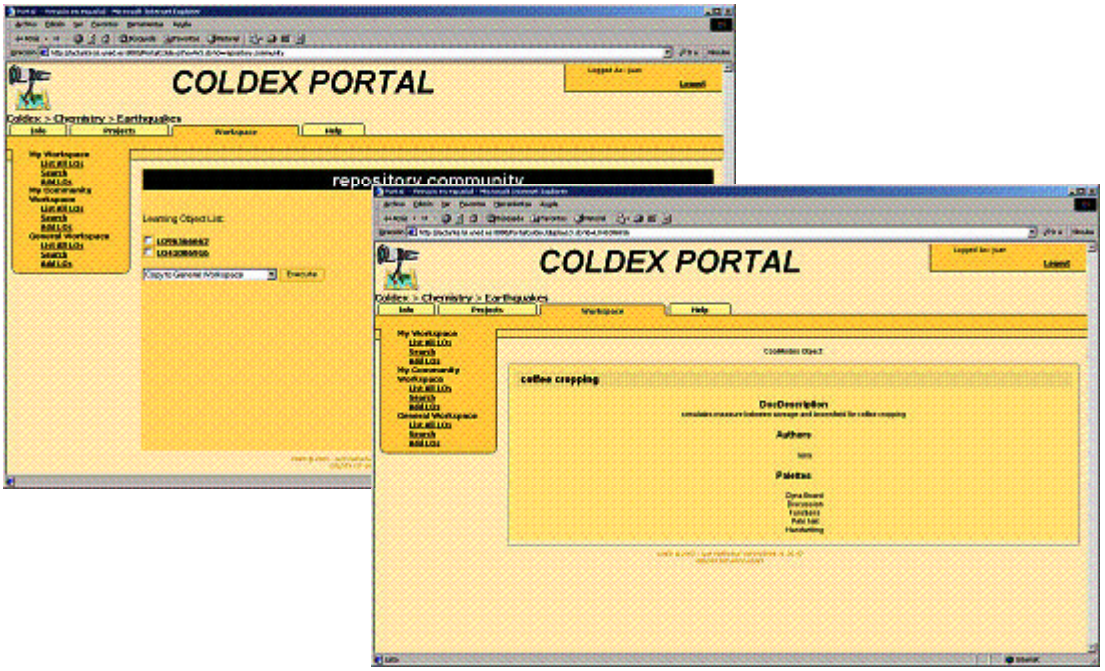


Figure 3: Listing Learning Objects

While working in the Private Workspace, a user can copy or move a Learning Object to the either the Community Workspace or to the General Workspace

4.2 Searching Learning Objects

Figure 4 shows how to search learning objects.

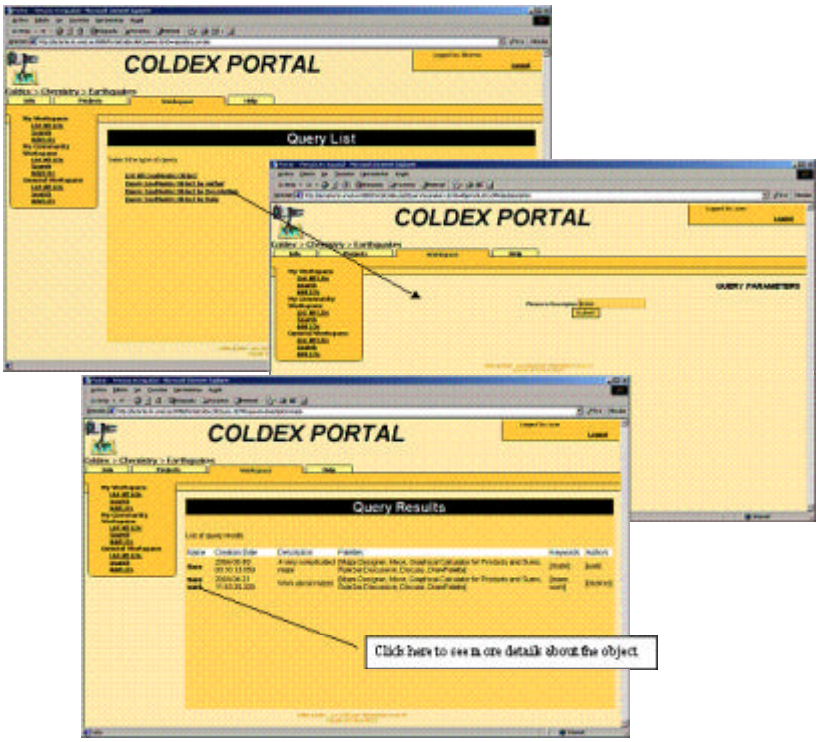


Figure 4: Searching Learning Objects

A list of queries is given for the *search* operation. In future versions of this prototype, there will be tools so that the users can compose their own queries. This way, they will take advantage of having an underlying ontology within the system. Nevertheless, the current queries are not closed: they are parameterised so that the user can customize them according to their preferences.

4.3 Adding Learning Objects

There is a few number of Learning Objects types stored in the LOR and available through the Coldex Portal. This number will increase as the fellow Coldex Groups send us their data descriptions.

Currently, the Portal shows a very simple user interface where the available learning objects can be selectet

4.3.1 Adding Learning Object Types

Including a **new type** of learning object into the system would require:

1. Designing a form for the Learning Object **contents**. This form is aimed to ask the user those data that are required to deploy and use the object (for instance, the width and height for displaying the object)
2. Designing a form for the (rest of the) Learning Object **metadata**, particularly, those which are specific to the new learning object (for instance, a CoolModesAsset would require the URL of the current database for CoolModes)

Both forms can be described in a number of ways, ranging from a formal DTD to a (*non ambiguous*) description of the different fields (or attributes) which will compose the forms. It is also very important to provide explanations defining clearly the relationship between Learning Object attributes and metadata, that is, which Learning Object attributes could allow computing what metadata. Furthermore, it would be very useful having **examples of the new Learning Object types** with the right values for their attributes

4.3.2 An Example of A New Learning Object Type

As an example, lets supose that we want to include a new type of Learning Object to reflect the *reports* the students will be writing about their learning experiences. Let be Astronomy the domain and capturing telescope images of the moon the overall project the students are working in. Then, for an activity about moon craters, each group of students involved in this project will be making a **report** as an HTML document. The new Learning Object type (**report**) could be described as follows (`attributes` are displayed in `typeface` while their *values* are written in *italics*):

- **content**
  - `name`: *moon craters report*
  - `file reference`:*Coldex/LearningObject/Reports/mooncraters.html*
  - `caption`: *Moon Crater Observations*

    - description: *Report on the telescope pics of the Moon Craters for the Astronomy Project made by the Group G001 at UPM*
- **other metadata**
    - author: *Group G001 at UPM*

      components:

        - *s01*
            - name: *Juan PÃ©rez*
            - login: *jperez*
        - *s02*
            - name: *RaÃºl LÃ³pez*
            - login: *rlopez*
    - creation date: ***2004.05.07 11:49:56.128*** (automatically generated?)
    - format: ***html*** (obtained from the file reference attribute)
    - file size: ***32KB*** (obtained from the file reference attribute)

The **content** form would include the attributes name, file reference, caption and description while the **other metadata** form would be composed by the attributes author (which can include individual components identified by name and login), creation date (which can be generated automatically), format and size, both obtained from the content attribute file reference)

The same example, now in the form of a DTD and two XML files would be:

- DTD for the content data:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT report (name, fileref, caption?, description?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT fileref (#PCDATA)>
<!ELEMENT caption (#PCDATA)>
<!ELEMENT description (#PCDATA)>
```

- report.xml (i.e., an example of content for the **report** LO):

```
<report>
  <name>moon craters report</name>

<fileref>Coldex/LearningObject/Reports/mooncraters.html</fileref
>
  <caption>Moon Crater Observations</caption>
  <description>Report on the telescope pics of the Moon
Craters for the Astronomy Project made by the Group G001 at
UPM</description>
</report>
```

- DTD for the rest of the report metadata:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT reportmdata (author, creation, format,
filesize)>
```

```
<!ELEMENT author (component+)>
<!ELEMENT component EMPTY>
<!ATTLIST component name NMTOKEN #IMPLIED>
<!ATTLIST component login NMTOKEN #REQUIRED>
<!ELEMENT format (#PCDATA)>
<!ELEMENT filesize (#PCDATA)>
```

- reportmdata.xml (i.e., an example of the rest of the metadata for the **report** object)

```
<reportmdata>
  <author>
    <component   name="Juan   PÃ©rez"   login="jperez"/>   ;
    <component   name="RaÃºl   LÃ³pez"   login="rlopez"/>   ;
  </author>
  <format>HTML</format>
  <filesize>32KB</filesize>
</reportmdata>
```

4.3.3 Adding Learning Objects Instances

As it is said before, the set of Learning Classes depends on the data made available to us. Presently, we have got a small number, which is bound to grow larger as soon as we get some feed-back. The interfaces are also subject to change in order to cope with the new objects and metadata to be added to the system. Figure 5 shows how to add an instance of an existing learning object

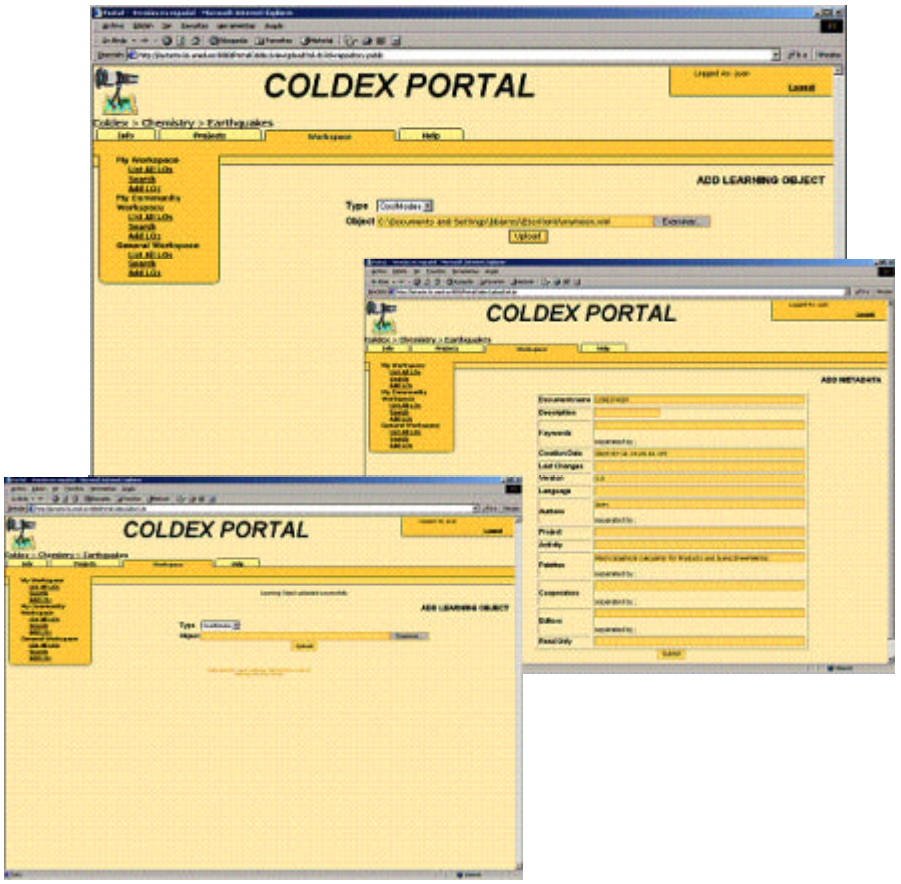


Figure 5: Adding Learning Objects