Using Digital Repositories to Support Project Based Work in On-Line Education

Carolina Castrillón Ojeda[*] Erwin Meza Vega [†]

Abstract

In order to take full advantage of the resources offered by content repositories, it is essential to combine the use of these resources with some teaching/learning strategies or methodologies, such as project based work. This methodology is based on the analysis of problems or situations that are found on the students' environment, and by using all the information and skills available, it is possible to develop activities that are motivating for both teachers and students and that may help to strengthen the teaching and learning process. This paper presents the integration of a tool for developing project based work activities with the required software tools to support content object creation, management and distribution, inside a Virtual Learning Environment. Firstly, an introduction to the project based work methodology along with its stages is presented, and then some details of the design and implementation of the project based work tool are explained. Secondly, the architecture for content object creation, management and distribution is presented, and also some implementation details are described. Finally, a strategy for integrating project based work tools and content management tools are presented.

Keywords*: Project Based Work, Digital Repositories, Sharable Content Objects, Virtual Learning Environments.*

Resumen

Con el fin de tomar ventaja de los recursos que ofrecen los repositorios de contenidos, es esencial combinar el uso de estos recursos con algunas estrategias o metodologías de enseñanza y aprendizaje, tales como trabajo por proyectos. Esta metodología se basa en el análisis de problemas o situaciones que se encuentran en el entorno del estudiante, y por medio del uso de la información y las habilidades disponibles, es posible desarrollar actividades que son motivadores para profesores y estudiantes, y que pueden fortalecer el proceso de enseñanza y aprendizaje. Este artículo presenta la integración ente una herramienta para desarrollar actividades de trabajo por proyectos con las herramientas software requeridas para llevar a cabo la creación, la administración y distribución de objetos de contenido en un Ambiente Virtual de Aprendizaje. Inicialmente se presenta una introducción a la metodología de trabajo por proyectos, y luego se describen algunos detalles de su diseño e implementación. En segundo lugar, se presenta la arquitectura para la creación, la administración y distribución de objetos de contenido, así como algunos detalles de su implementación. Finalmente, se presentan una estrategia para integrar la herramienta de trabajo por proyectos con las herramientas de administración de contenidos.

Palabras clave*: Trabajo por proyectos, Repositorios Digitales, Objetos de Contenido Compartible, Ambientes Virtuales de Aprendizaje.*

[*] Universidad del Cauca, Calle 5 Nro 4-70, Popayán, Colombia.
[†] Universidad del Cauca, Calle 5 Nro 4-70. Popayán, Colombia.

1 Introduction

On-Line education is seen as an alternative for developing distributed and distance teaching/learning experiences, which arise from necessity of extending the coverage and/or the variety of the academic programs offered by universities. In order to obtain effective results from the possibilities offered by the Information and Communication Tools (ICT), most of the universities have started to implement some strategies of integrating ICT into their academic programs. Nevertheless, this integration is not a simple process of taking contents and putting them into the web, and allowing the students and the teachers to interact with each other through some communication tools. Any attempt of integrating ICT into education should start from the analysis and planning of activities related to the teaching/learning experiences that are intended to be developed, in order to create educational scenarios that fit to the necessities and opportunities of the students, and also considering the educational environment and the available resources. In this direction, efforts must be focused on building meaningful and autonomous learning, and also leveraging the “know how”, “know why”, and “how to learn” approaches instead of the common “how to use” approach .

With the aim of integrating both teaching/learning methodologies and ICT, the GENTE group (Spanish initials for Group of Study and Research on Technologies and Education) started a research project in years 2001-2002, in which a Virtual Learning Environment (VLE) called Aula Virtual was developed. This Virtual Learning Environment offered the common ICT tools, such as e-mail, chat rooms and forums, as well as content, calendar and link management. Aula Virtual also implemented a project based work tool, which was conceived as tool for organizing teaching activities, in a way that these activities have a real meaning for the students. By using both ICT tools and project based work, students and teachers of several courses (more than 100 courses from 2002 to 2005) have developed supplementary activities in order to enrich the traditional method of teaching and learning used on the Universidad Industrial de Santander.

The experience obtained from all these courses has shown a very important issue that was not taken into account in first version of Aula Virtual: By using ICT tools, students and teachers create and share a great amount of information, and most of the resources can be reused on later experiences. Resources can include student and teacher notes and content, and documents obtained from the development of project based activities. In order to solve this issue, a second version of Aula Virtual and developed in years 2005-2006 . This new release includes the required functionality to create and display SCORM-compliant content objects, and also the means for storing, searching and managing content objects stored into a Content Repository, also developed during this second implementation of Aula Virtual.

The rest of this paper is divided into three major sections: the first section presents a brief introduction to project based work and its role in on-line education, afterwards some design and implementation details of project management tool are described. The second section explains some technical details about the current design and implementation of the Aula Virtual modules that are related to content object creation and management, including some details about the content repository associated with Aula Virtual. The third section describes the strategy that was established to integrate both project management tools and content management tools. Finally some conclusions and future work hints are presented.

2 Using Project Based Work to Support Teaching and Learning

Project based work is conceptualized as dynamic process of organizing teaching and learning, by means of a set of activities that have a real significance for students . From this perspective, projects can be conceived as an individual or group, integrated-to-context free-choice plans, which aim to develop some real-life activities that are interesting for both teachers and students. All projects share the following characteristics: (1) a project must promote the reasoning and problem-solving abilities to real situations; (2) information is not transmitted by itself, it has to be searched and classified; (3) the learning activities are performed on a real environment; and (4) teaching is focused on problems, prior to any principle or theory.

Nowadays, project based work was adapted to new pedagogical models such as competence-based learning. Thus, projects are focused on building one or more competences, starting from a real-life problem that must be solved by using several sources of information and areas of knowledge. In order to integrate all the information, resources and skills available, some predefined sequence of phases must be followed while developing the project. These phases can be summarized as follows :

- *Learning path definition*: The Learning path is a guide that defines the basic project information and characteristics, such as project name, academic program, situation to be analyzed and solved, projected complexity level, assessment methodology, and some associated resources. All this information allows classifying the projects into applied projects, or research projects. Applied projects use different theories and concepts in order to solve a specific problem. By the other hand, research projects seek to obtain new knowledge.
- *Prior knowledge identification*: Students and teachers must identify their competences and prior knowledge. The result from this analysis motivates both students and teachers to start the development of the new project to obtain new knowledge or develop new competences.
- *Project layout*: This phase aims to define the basic policies that must be followed by students and teachers while developing the project. These policies include defining the intended products of the project development, which can be classified into project products (specific solutions), or knowledge products (related to building new knowledge). Also, basic communication and interaction policies must be established, such as individual or team work, accomplishment of assigned jobs and activities, respect for the others, and so on.
- *Initial project setup*: After defining all previous items, students and teachers engage the task of defining the problem. This includes diagnose, problem statement, objectives and justification, theoretical background, schedule, resources involved, and evaluation parameters. According to the complexity of the project, some of these aspects may be skipped.
- *Project development/execution*: In this phase, students carry out the development of the project, according to the previously defined schedule and rules. Teachers must take a guiding role through the execution of the project stages, in order to allow students to build their own competences and new knowledge. Guiding includes facilitating the required resources (i.e. links to content, web pages and tools, hints and motivating messages) that students may need. By the other hand, students may interact with teachers and other students in order to accomplish the defined activities, and they also can look for additional resources. Additionally, both teachers and students may build new resources that can support the development of the project.
- *Evaluation of the project development*: In order to identify possible flaws of the project development process, teachers and students must evaluate each stage of the project. Flaws can include non-planned activities, inadequate teacher assistance, delays, and lack of information, amongst others. The result of the evaluation can prevent those flaws from happening again.
- *Evaluation of the individual or team achievements:* Also, it is necessary to evaluate each student or work team, in order to establish the level of achievement of the proposed goals. This evaluation process can be performed either while developing the project or at the end of the project.

2.1 Project Based Work in On-Line Education

There are several experiences of using project based work in on-line education , , . From all these experiences, it is visible that project based work and ICT can increase the quality, variety and value of teaching and learning activities. It encourages individuals or groups to carry out in-depth, interesting projects that make use of several sources of information, and at the same time, ICT facilitate the creation, acquisition and organization of the information, as well as the possibility of communicating among teachers and students. As stated in , "Students of all ages have the knowledge, skills, and interest to work on 'authentic' learning tasks —those that involve substantive, real-world problems. Projects into which technology is integrated often involve authentic work".

One important issue about using project based work in on-line education is that it becomes learner centred. This means that students play a significant role in selecting the nature, goals and scope of the projects they develop. Thus, teachers must encourage students to understand what they are doing, why developing projects is important, and how they will be assessed and evaluated during the process. Certainly, students become more motivated with the project development, and they are inherently motivated to acquire new competences or building new knowledge .

2.2 Designing and Implementing a Project-Based Work Tool for Aula Virtual

Starting from the experience obtained from the first implementation of a project based work tool for Aula Virtual 1.0, the research project "Prototype for the Development of Specialization Programs Based on Virtual Learning Environments" proposed a project-based work tool for supporting the development of specialization programs. The general requirements of the planned project-based work tool were:

- Each project is built upon a series of activities, that can be classified into: (1) general activities, assigned by teachers to all students, (2) individual and/or team activities, also assigned by teachers, and (2) personal activities, assigned by teachers and students. Activities can be developed concurrently.
- Every activity must have the required mechanisms to allow communicating teachers and students, e-mail, chat rooms and forums were considered as the basic mechanisms.
- The result of each activity consists of a set of resources (i.e. texts, slides, images, multimedia, and diagrams) that must be stored in a way that they can visualized by the teachers, and they should be available to be reused on later projects. Likewise, teachers must facilitate the access to several resources, such as guidelines, work documents, tutorials or manuals.
- Teachers must have some assessment tools, which should be used to follow the individual / team progress through the development of the project. Also, teachers must provide continuous feedback to students / teams about the results obtained on the development of each activity.
- And lastly, this tool must be integrated into Aula Virtual, so it could be used on any course inside this Virtual Learning Environment.

Also, some issues were considered. These are briefly explained below:

- Projects can be very complex. For this reason, teachers and students are not allowed to develop concurrent projects inside a course. However, they can participate in other courses projects.
- By default, all projects are developed individually. Defining work teams is a choice taken by teachers and students while defining the project layout. The project based work tool must be flexible enough to allow developing projects in which participate both individuals and work teams.

- While developing project activities, any resource associated to a specific activity is visible only to teachers and students of the same course. In work team activities, resources can only be accessed by teachers and members of each team. Once the project has ended, all those resources can be made public, so that they can be reused by teachers and students from any course.
- Projects can last hours, weeks or months. The duration of the project is determined by the duration of their associated activities, which can depend on the complexity of the project.

2.2.1 Project Based Work Tool General Use-Cases Diagrams

The following subsections describe some general use-cases diagrams that were obtained while designing the project based work tool.

Project management

The use-cases diagram in Fig. 1 shows the basic use-cases for project management. Basically, teachers are responsible of creating, modifying and deleting projects inside any Aula Virtual course. Management also includes defining some initial activities in order to set up the project development, and defining work teams, if necessary.

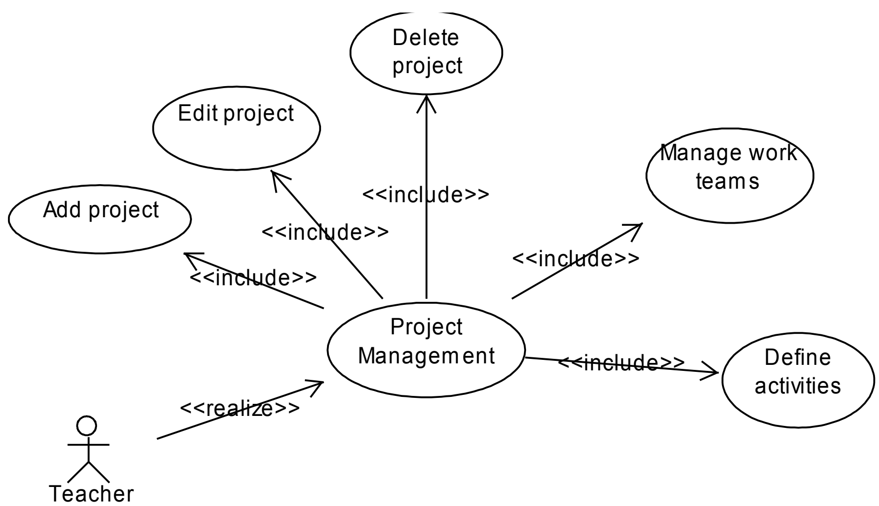

Fig. 1. Project Management use-cases diagram

Adding a new project requires defining some general parameters, such as project name, planned start and end date, general and specific objectives, location (context) of the project, and some initial resources that can be used as a theoretical background.

Also, as stated on the general requirements, activities can be classified into:

- *General activities*: also called project activities, are all those that must be executed by every participant of the project development process (i.e. problem definition and analysis, resource gathering, chat sessions, and so on).
- Team and/or individual activities: are intended to develop specific team/individual activities, such as suggesting solution strategies for the defined problem, or defining problem solution scenarios. These activities can also be used to support those individuals/teams that have difficulties while developing the project.

- Personal activities: These activities allow creating a personal calendar that can be accessed only by its creator (either teacher or student).

Project development

Once the Project has been defined, teachers and students carry out its development. This includes developing the activities defined by the teacher, and also defining new activities, if necessary (see Fig. 2).

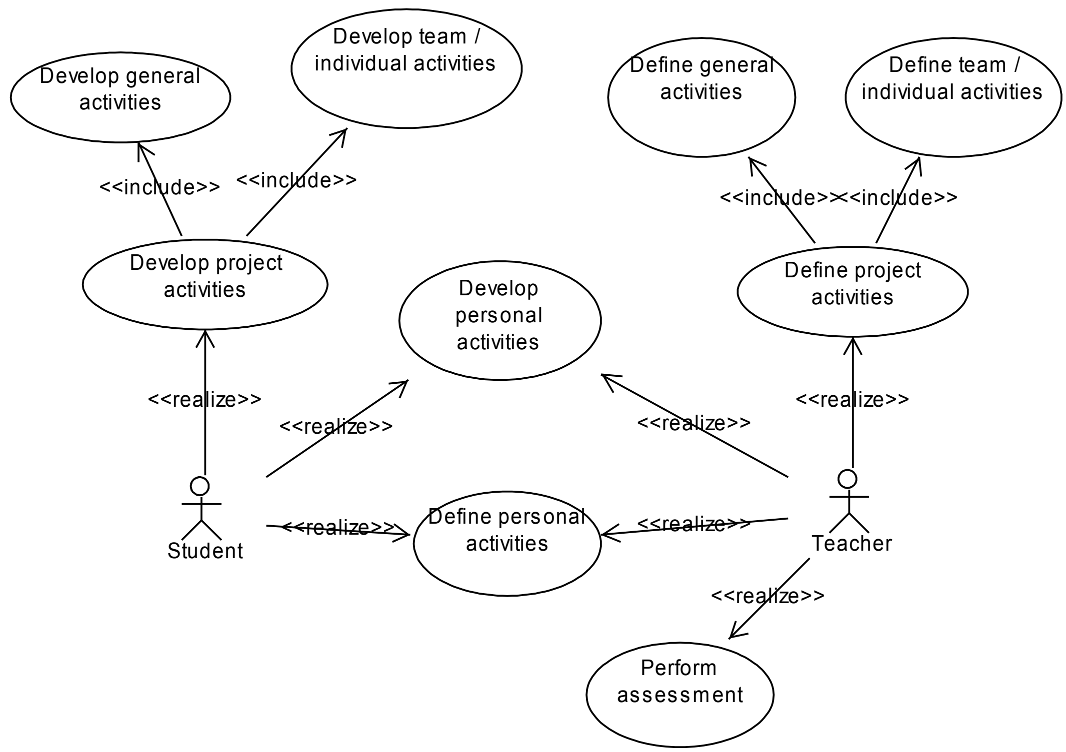

Fig. 2. Project development use-cases diagram

Each general or individual/team activity must be associated to a specific goal of the project. This allows defining some indicators that will determine if the planned objectives were fulfilled. By the other hand, teachers can perform assessment during the development of the defined activities by motivating students, facilitating resources and guiding the overall development of the planned activities.

Activity development

Developing a general or individual/team activity requires the support from ICT tools, such as content management, calendars, chat rooms, and discussion forums. Therefore, each activity has an associated forum, in which students and teachers can interact with each other, in order to make suggestions, define some supplemental activities, share resources, and solve issues that can take place. Additionally, teachers and students can associate relevant resources to the activity, such as documents, links to external pages, multimedia clips and so on. For this to be possible, teachers and students must have at hand content management tools that allow creating, searching and locating content resources. The activity development use-cases diagram is shown in Fig. 3.

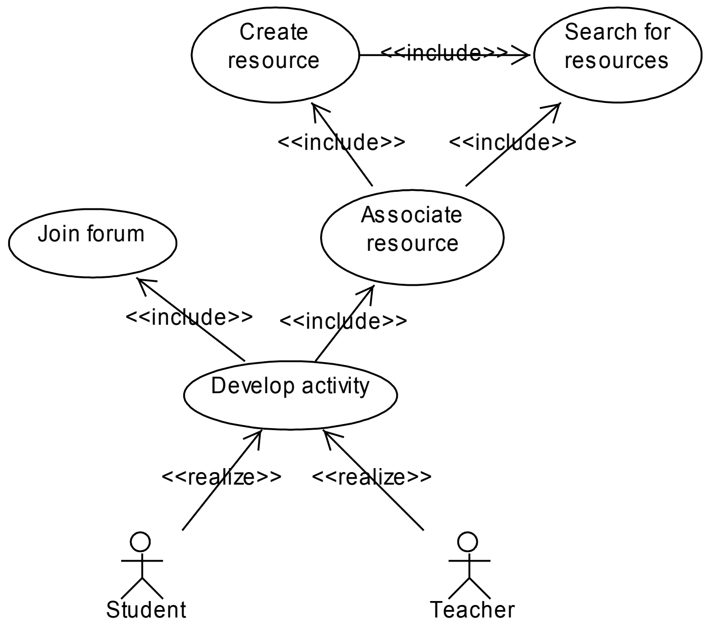

Fig. 3. Activity development use-cases diagram

Depending on the complexity of the activities, the creation of a complex resource can require searching for simple content resources (such as images, multimedia clips and web pages). This feature must be supported by content management tools, which should allow reusing existing resources by means of search, content assembly and content storage and distribution utilities.

2.2.2 Implementation of the Project Based Work Tool

The project based work tool was implemented as an additional Aula Virtual 2.0 module that makes use of several services provided by this Virtual Learning Environment, such as Interface services, Content Management services and General Run-Time Services, amongst others. Interface services are related to how the information is displayed inside the Virtual Learning Environment, and it includes the basic primitives for creating menus, forms and so on. Content Management services offer the basic functionalities for building and storing content resources, and Run-Time services handle the requests generated from the user interface, and they perform the required logic operations, which include session handling, data retrieval and processing and passing callbacks to other modules.

The general view of the project based work module shows how user events trigger the execution of several Aula Virtual Modules (see Fig. 4). In this case, a request was made to the project management module, that uses the Run-Time Environment to retrieve the session data, along with the submitted data from the user, and it uses several libraries to perform the required action. Also, modules can communicate with each other through a module management API, which allows sending requests back and forth through modules, in form of callback functions (i.e. Process form data, retrieve and draw current week calendar, build a menu, get some records from the database and so on).

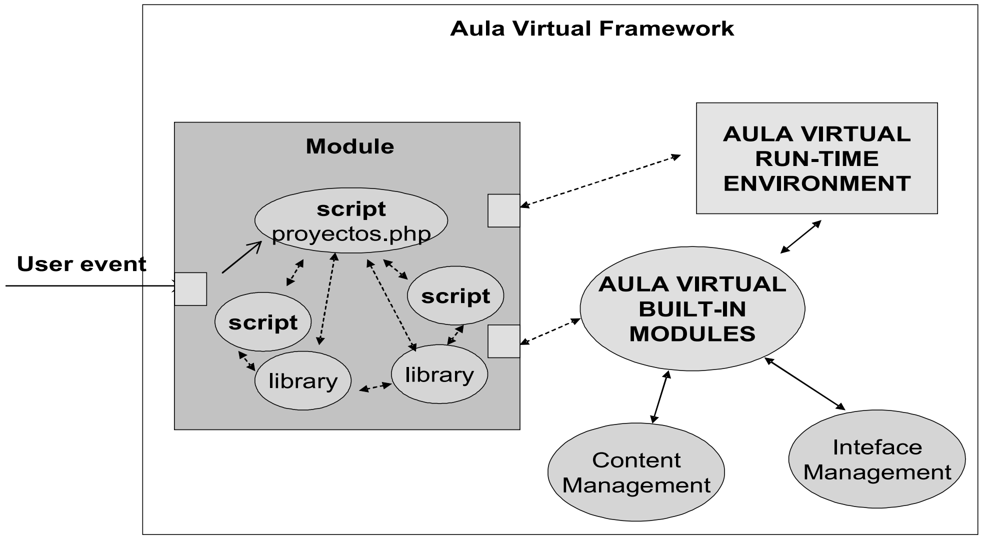

Fig. 4. General view of the proyect based work module

As mentioned before, Aula Virtual interface management module is responsible for building the final user interface. To do this task, this module invokes the appropriate callback functions for each registered module. There are two basic callback functions: module_get_menu_option, and module_get_data. For instance, if project management module is active, the interface management module invokes the proyectos_get_menu_option and proyectos_get_data callback functions. Additionally, the module_get_menu_option callback function is called for each registered module. As shown in Fig. 5, the registered modules are actividades (activities), contenidos (contents), proyectos (projects, which is also active), estadisticas (statistics), personalizar (preferences), recursos (resource management), enlaces (external links), ayuda (help).

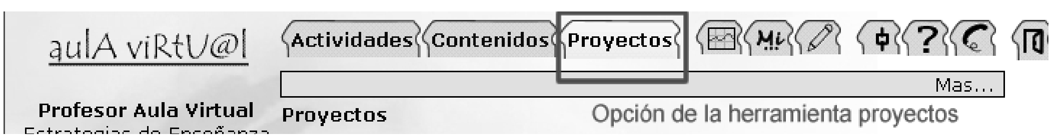

Fig. 5. Project module menu option inside Aula Virtual user interface

New project addition

The project addition interface allows teachers to specify the required parameters to start a project, such as name, description, start and end dates (see Fig. 6). The project management module checks that there are no other projects that overlap the new project start and end dates, and then the new project is created into the current course.

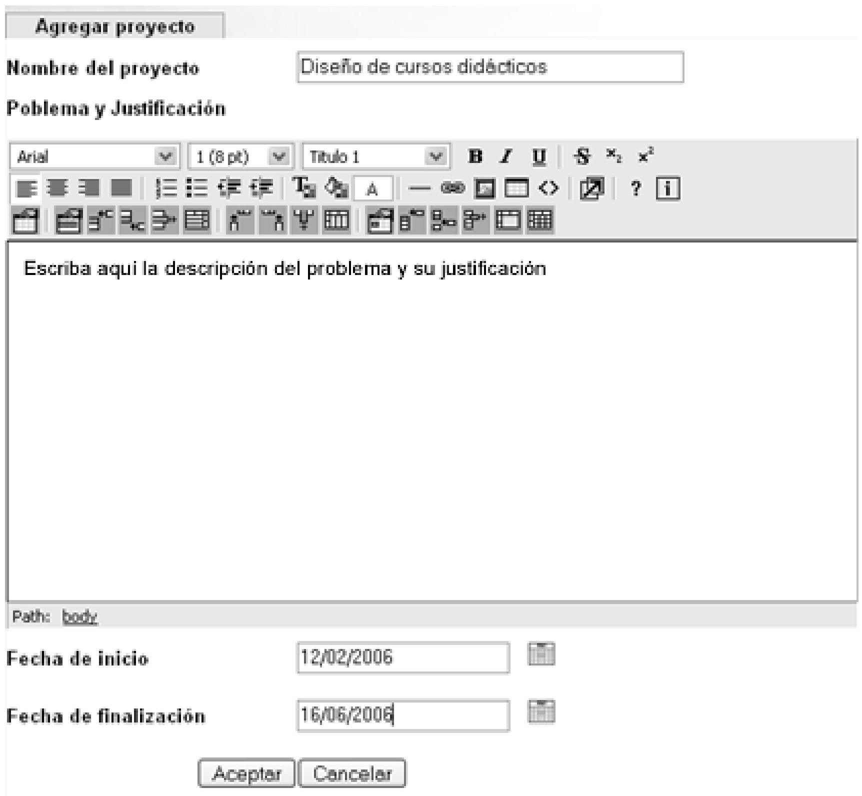

Fig. 6. Project addition interface

Project management / development interface

This interface allows managing and / or developing the project, according to the role of the current user (See Fig. 7). Basic options are explained below:

- *Inicio (Start)*: Presents the basic project information: Project name, description, start and end dates. This information is edited by the teacher.
- *Descripción (Description)*: It shows other general information, such as project context, location, related student competences and theoretical background. This information is also edited by the teacher.
- *Objetivos (Objectives):* It describes the general and specific objectives of the project. Teachers can edit the general objective, as well as add/edit/delete specific objectives
- *Grupos (Work teams):* By using this option, teachers can set up the work teams, if project requires them. Every student can only join a work team.
- *Actividades (Activities):* This option can be used to define new activities (general, individual/team or personal). For each general and individual/team activity, a discussion forum is automatically created.
- *Recursos (Resources):* While developing the project activities, teachers and students use this option to create, search and associate new resources to a specific activity. These resources can be text documents, web pages, slides, and it is also possible to associate SCORM content objects to an activity. This feature is explained later in this paper.
- *Indicadores (Achievement indicators):* Teachers can define a set of achievement indicators, which will be used as a template to evaluating the level of achievement of the project's specific objectives.

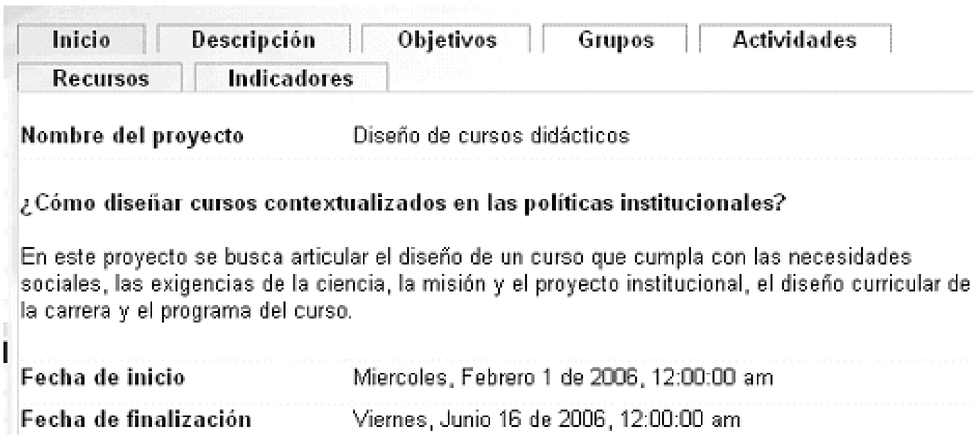

Fig. 7. Project management / development interface

3 Using Content Repositories to Support On-Line Learning Activities

On-Line learning activities that are developed inside a Virtual Learning Environment often require creating, storing and managing resources. In this way, developing tools that support content management is an important issue that must be taken into account. In order to address this issue, specifications and reference models such as the Shareable Content Object Reference Model (SCORM) , the IMS Digital Repositories Interoperability (DRI) and the Content Object Repository Discovery and Resolution Architecture (CORDRA) have been developed. These initiatives provide the basic guidelines for building reusable content objects and the Virtual Learning Environments that supports them (SCORM) , the means required to interoperate Content Object Repositories (IMS DRI) , and the foundations of content discovery through federated search (CORDRA) .

3.1 An Architecture to Support Resource Management

The analysis of the initiatives mentioned before was one of the first steps towards the creation of the second version of Aula Virtual. In this way, the research project “Platform for the Exchange of Reusable Educational Content among Virtual Learning Environments” (in which Aula Virtual 2.0 was designed and built) established the necessary relationships between Virtual Learning Environments and Content Object Repositories, in order to provide a set of software tools and Web Services for building, storing and managing SCORM-compliant learning objects and their associated metadata, and also searching and discovering learning objects stored in several content object repositories. Later on, a web services based architecture for the platform was proposed (see Fig. 8**)** . This architecture, which is based on SCORM and the IMS DRI specification , provides the basic building blocks to allow creating, storing, managing and searching SCORM-based content objects that are intended to be used in on-line learning activities.

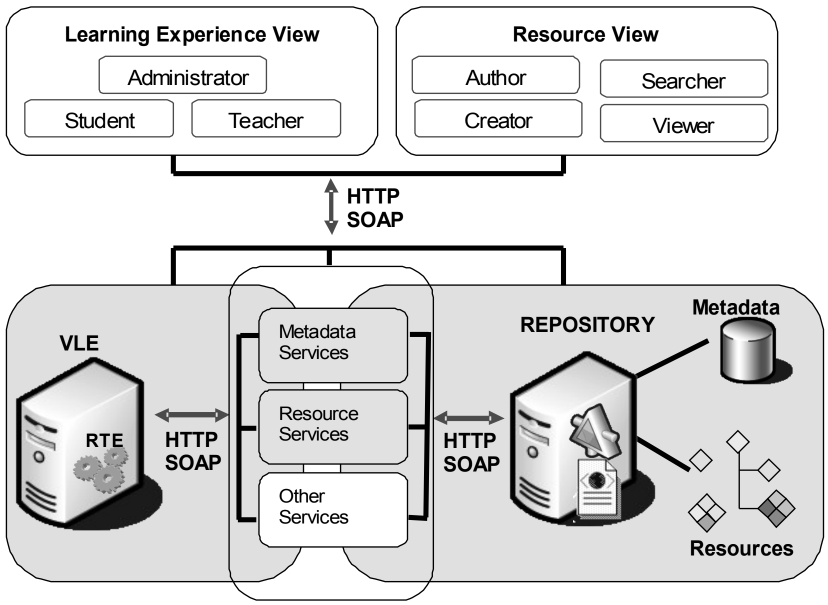

Fig. 8. Virtual Learning Environments and Digital repositories interoperability architecture

3.1.1 Users of the Proposed Architecture

Users are the most important components of the proposed architecture. They can build, store, manage, search and reuse content resources (and their associated metadata) on several learning activities. In order to distinguish the different roles that users play on the architecture, they considered from two different views:

- *Learning experience view*: From this view, users are classified into administrators, teachers and students. Administrators are responsible of setting up the learning environment (creating users and courses, granting access to the virtual learning environment and the content repository, registering students and teachers, among other tasks). Teachers set up learning experiences, prepare contents, store and search resources inside the repository. Students develop learning experiences, at the time that create, search and visualize contents.
- *Resource view*: Regardless of their role on a learning experience, users can perform several resource related tasks, and they can be classified into Creators, Authors, Searchers and Viewers. Creators build the basic blocks of content, such as text pages, images, multimedia clips and animations, and store those contents into the repository. Authors use the resources developed by Creators, in order to build more complex units of instruction. According to SCORM, these units are Assets, Shareable Content Objects and Content Aggregations . Searchers perform queries to the repository, in order to locate specific resources. Finally, Viewers visualize the content resources stored into the repository.

It is important to note that users are very dynamic: They can act as students and teachers on several courses, at the time that they create, search and visualize resources as they need to. Also, Authors often need to act as Searchers and Viewers, in order to locate possible reusable content. Searchers can visualize the resources they are looking for, and so on.

3.1.2 The Role of Virtual Learning Environments

Apart from their common duties, Virtual Learning Environments must provide enough tools to make content management tasks easy. These tools can vary from system to system, but they must be based on some standard, reference model or specification. For Aula Virtual, the SCORM reference model was selected, and some extra components were added in order to allow using content repositories. The most important tools provided by the VLE are:

- *SCORM-Compliant authoring tools:* By using WYSIWYG (What-You-See-Is-What-You-Get) editors, users can create from simple text pages to pages with tables, images, graphics and multimedia elements, as well as SCORM-Compliant content objects. Once created, these resources can be stored inside the Content Repository or the VLE itself.
- *Repository management:* Users can store and access resources from several repositories. For this to be possible, VLEs must facilitate the registration, communication and use of services from different repositories, according to a pre-defined protocol.
- *SCORM Run*-Time Environment: This tool is intended to take advantage of the sequencing and navigation features defined in the SCORM reference model.

3.1.3 Content Repositories

As well as users and virtual learning environments, repositories play a significant role in the proposed architecture. They must provide the basic services for content storage, management and searching . The proposed architecture uses two different approaches: web services interfaces and regular web interfaces, in order to target two different types of users: human users, that can access directly to the repository through the web interface, and other application, that make use of the web services interface to access the provided services.

Web services interface

The web services interface provides the means required to achieve the interoperability between applications. These services are based on the IMS DRI Specification , and they offer a set of methods that can be called through SOAP requests. So far now, the services offered by the repositories can be classified into general, metadata and resource services:

- *General Services:* Allow the users to register and to establish the communication with the content repositories. Subscribe service allows registering new user into a repository, and the Connection Test service can be used to check the connection status with the repository.
- *Metadata services*: This set of services allows the users to perform several metadata related tasks, such creating and submitting metadata to a repository. The Metadata Load service allows the users to submit a resource metadata record into the repository. Metadata must be structured in form of a XML document, according to metadata specifications such as IEEE LOM or Dublin Core . The Metadata download service can be used to download a resource metadata, which is built on-the-fly from the relational database model and sent to the user as a XML document. The Metadata Search service facilitates searching content objects, according to a specified search pattern. Finally, the Metadata Schema Information Service can be used to determine the different metadata schemas currently supported by the repository.
- *Resource Services:* These services allow loading and downloading resources from the repository. The Load Resource service is based on the Submit/Store function defined by the IMS DRI specification, and the Download Resource allows the users to download a resource from the repository through a HTTP connection.

Web Interface

The aim of the web interface is to offer a simple way for users of having direct access to the services provided by the repository. This interface is also a method for testing the performance of the web services interface, in order to improve its design. The main components of the web interface are:

- *Subscribe:* This is the first step that users must perform in order to use the services of the repository. The subscribe interface is a simple web form, that once filled and submitted back to the server, makes use of the subscribe web service to obtain a use key. User keys are the basic authentication credentials that must be used to access any repository service.
- *Resource management*: This interface allows the users to manage their resources within the repository. Once logged by means of the key, users have the possibility of adding new resources, editing content object metadata, searching, downloading and visualizing resources.
- *Metadata editing*: One of the most powerful tools offered by the repository web interface is the metadata editing tool. This tool allows the users to load/edit/import/export the metadata of a given learning object. To allow a high level of interoperability, the metadata editing interface does not rely in a specific metadata schema, instead any metadata schema, such as IEEE LOM or Dublin Core can be used to describe the resources. This approach was selected from the development of another repository implementation, which proposes a model for integrating several Latin-American content and metadata repositories into a network called EduLatin .

3.2 Implementing the Proposed Architecture

Once defined, a prototype for each component of the architecture was implemented, in order to validate the design . These components were developed using free software, with Linux as the operating system, PHP and Java as programming languages and MySQL as the data persistence mechanism. The final result is a Virtual Learning Environment called Aula Virtual 2.0 and a standards-based Content Repository . These tools are also being tested, with the aim of integrating this solution into Edulatin Network . Next sections briefly describe some details of the currently developed implementation of the architecture.

3.2.1 Repository Implementation

The repository implementation developed so far now follows at a certain degree the IMS DRI specification , and it allows storing resources and metadata as well as searching for stored resources. Once subscribed, users can use both the web interface and the web services interface, according to their individual possibilities. The implementation of these interfaces is briefly explained below.

Web Services Interface

The web services interface is the core component of the entire architecture. It provides the required abstraction to allow the interoperability between several repository and VLE implementations. As mentioned before, these services are implemented as a set of methods that can be invoked through SOAP request. Fig. 9 shows how the services can be invoked from a client application.

```
$client = new SoapClient($CONFIG->repository_wsdl);
set_time_limit(0);
$result = $client->subscribe(Array("name" => $name),
                                "email" => $email),
                                "url" => $url));
$key = $result->suscribeResult; //get web service response
```

Fig. 9. Invocation of a repository web service

In order to achieve a higher level of interoperability, all SOAP requests receive and return simple-typed parameters, such as strings, numbers and simple objects. Consequently, complex results are returned as XML documents that must be parsed on the client side. This approach differs from current implementations that propose a specific protocol for sending and retrieving data between the client and the web services interface.

The core services offered by the web service interface are summarized below:

- *Connection test:* This service acts like the *ping* network utility: it sends a token (string) to the repository and waits for a response. In there is no response from the repository at a specified time interval, the user (application) must assume that the connection with the repository is not possible at the time.
- *Subscribe*: The user (application) sends a request to a repository, specifying a name, an email address and a web page URL. With this data, the repository creates a user key, which must be added on subsequent requests (see Fig. 9 for some details).
- *Get reference model list:* Before any attempt of storing a resource or metadata into the repository, the get reference model list method must be used, in order to retrieve a list of the supported reference models that define the valid schemas for a particular repository. This method uses the key obtained from the subscribe method, and returns a XML document, in which reference models supported by the repository are described.
- *Get component list:* In conjunction with the get reference model list method, this method must be used to retrieve a list of the supported components within a defined reference model (for SCORM , assets, SCOs, and content aggregations are currently supported.
- *Add resource:* The client must provide the user key, a reference model and component key, the resource name, description and primary language, and also a URL in which the resource can be located. With this information, the repository "loads" (transfers) the resource from the specified URL into the repository and creates a new metadata record, according to the reference model and component defined by the user. Once added, the repository returns a resource unique key, and a URL which can be used the access the resource directly.
- *Load metadata:* This method allows the users to load a metadata record into the repository for an existing resource. The metadata must conform to the schema defined for the resource, which is determined by the combination of the reference model and the component specified by the user for that resource. For SCORM resources, IEEE LOM must be used. The input parameters are the user key, the resource key and the metadata (in form of a XML document).
- *Download metadata:* By using this method, users can retrieve the metadata record stored in the repository for a given resource. The input parameters are the user key and object identifier, and the output is a string which containing the resource metadata as a XML document.
- *Metadata query:* In order to locate the resources stored into a specified repository, users can perform query requests to the metadata records of the existing resources, The implementation of this method follows the search method used by the Google Search Web API , in which the search pattern can include some special commands, such as the type of the resource (given as the component key), and a date limit (which limits the results only to those the resources added later to the specified date). As an example, a query pattern can be the specified as follows: "development type:SCO dateAdded:05/02/2005"; in this case the repository will search for Shareable Content Objects (SCOs) that contain the word "development" into their metadata, and were added later than February 5th, 2005.

Web interface

The repository web interface was developed with two goals in mind: testing the web services interface, and also allowing direct access to the repository. In this way, a web interface for each defined service was implemented into a web application, which is available for public use . As an example, the search interface is shown in Fig. 10. This interface can be used to obtain the user key, which is required to perform any operation inside the repository.

Fig. 10. Repository subscribe web interface

3.2.2 Virtual Learning Environment Implementation

The current VLE implementation provides administrators, students and teaches a simple but powerful interface for repositories-supported content management. Users can create new resources by starting from scratch or by using the available services for content location and reuse. Also, users can submit and store, search and visualize all the available resources on the content repository. A brief description of each VLE component is presented below.

Repository Management

The repository management component acts like a local registry, in which the information related to content repositories is stored. This information must include the registration data of each one of the repositories, and the location of their web services and web interface, which is required to establish a connection with them (see Fig. 11). This component is managed by the administrator of the VLE, and it should be configured in order to allow students and teachers selecting which of repositories they want to subscribe to. This approach promotes the use and creation of several dedicated repositories, all of them connected through the same services interface.

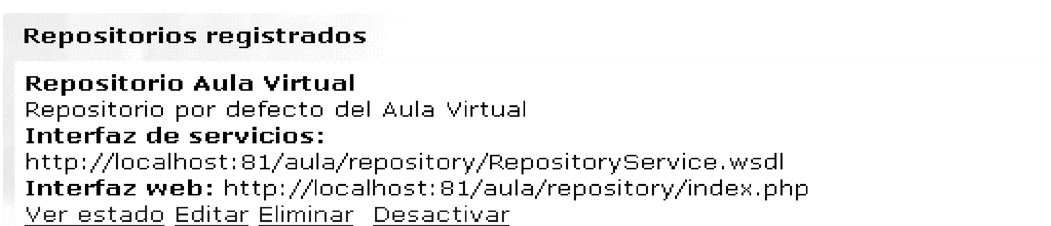

Fig. 11. Repository management administrator interface

Once the administrator has configured the access to several repositories, they can be accessed by both teachers and students. For this to be possible, users must subscribe to the content repositories they want, in order to obtain a repository user key. This key will be used on every request made to the repositories. Fig. 12 shows the repository management interface for a teacher/student: in this case, user is registered into the first repository, and he/she has not subscribed into the second one.

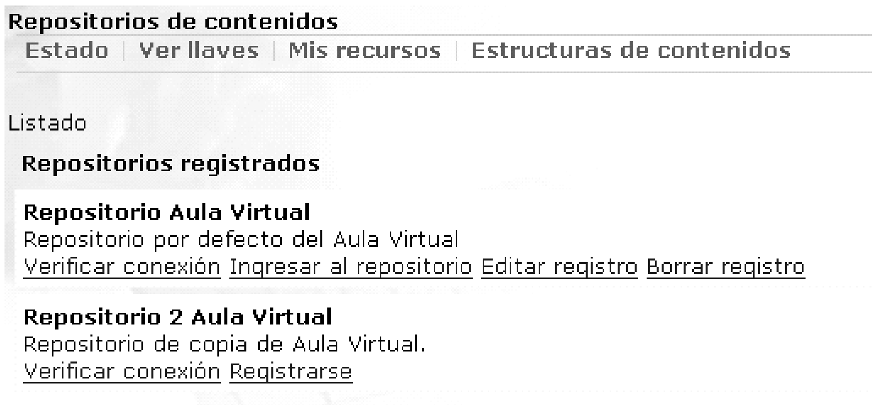

Fig. 12. Repository management teacher and student interface

Content Object search tool

This straightforward query interface allows the users searching and locating content objects stored on the repositories they are subscribed to. As shown on Fig. 13, users can submit a query on a specific repository, or submitting the query all of them. When users click the submit button ("Ver Recursos" button in Fig. 13), the VLE repository engine takes care of sending the search requests to the selected repositories, and later retrieving the matching records and displaying them to the user.

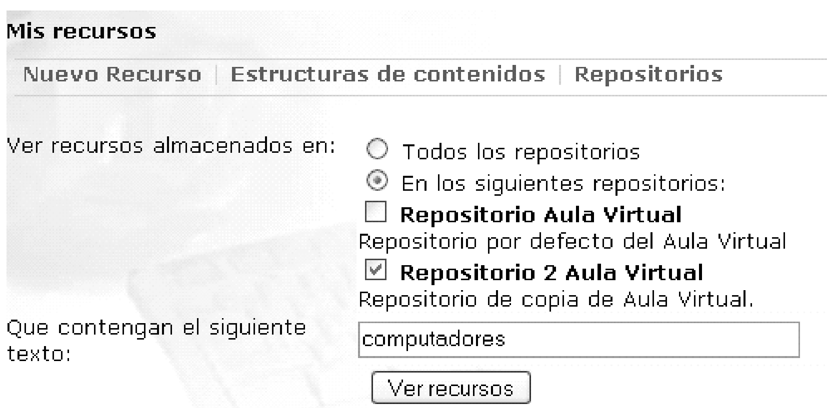

Fig. 13. Content search tool

Content Authoring tool

This tool provides simple web interfaces for creating new content objects (SCO, assets and content aggregations) that follow the specifications of the SCORM reference model as closely as possible. Fig. 14 shows the content authoring interface, in which different resources can be added, such as web pages, images, text, Assets, SCOs and even SCORM Content Aggregations. The authoring interface allows adding, editing, moving and deleting these resources at any time.

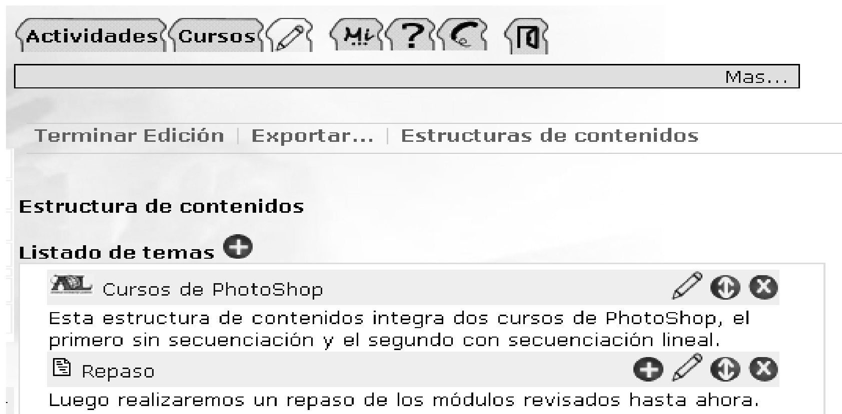

Fig. 14. Content authoring tool interface

To create a content structure, authors follow a simple book metaphor: the book index is presented, and authors can add new content (items) to that index. The content structure shown on Fig. 14 has two elements: Curso de PhotoShop (PhotoShop course), and Repaso (Review). The first is a SCORM content aggregation, and the latter is a simple resource (SCORM asset). Any item can be edited, moved to another place into the content structure, or deleted.

Once created, content structures can be packaged and stored into the content repository. These structures (known as content organizations in SCORM) are required to be packaged into a ZIP file, and then submitted to the repository through its web services interface. The Virtual Learning Environment makes this task easy, by compressing automatically the contents into a ZIP file, and then displaying a simple web form that the user must fill. The VLE submits the ZIP file to the repository, and according to the data provided by the users, it stores the new resource in form of an asset, SCO or Content Aggregation.

Fig. 15 shows the web interface for adding a new resource into a content repository, once the content structure has been exported to a ZIP file. Because it is possible that content repositories support several content and metadata specifications (such as SCORM or Dublin Core), users must specify which reference model was used (Modelo de referencia). Then, users select the resource type from that model (Tipo de recurso), the resource name (Nombre), description (Descripción), primary language (Lenguaje), ZIP file location, and start file (Archivo de inicio). Once this data is filled, the VLE communicates with the selected content repository, and submits the ZIP file and the description information from the web form. The content repository stores the ZIP file, and creates a new metadata record from the information provided by the user through the web form, and finally it returns the new resource unique identifier to the VLE.

Fig. 15. Web form for adding a resource into a content repository

4 Putting the Pieces Together

With all these tools at hand, teachers and students can develop project based work inside Aula Virtual, by combining ICT tools and the content management facilities provided by this Virtual Learning Environment and the Content Repository. Some application scenarios of project based work are presented below.

4.1 General Use Cases View

The diagram in Fig. 16 presents the main actors that use the provided software tools.

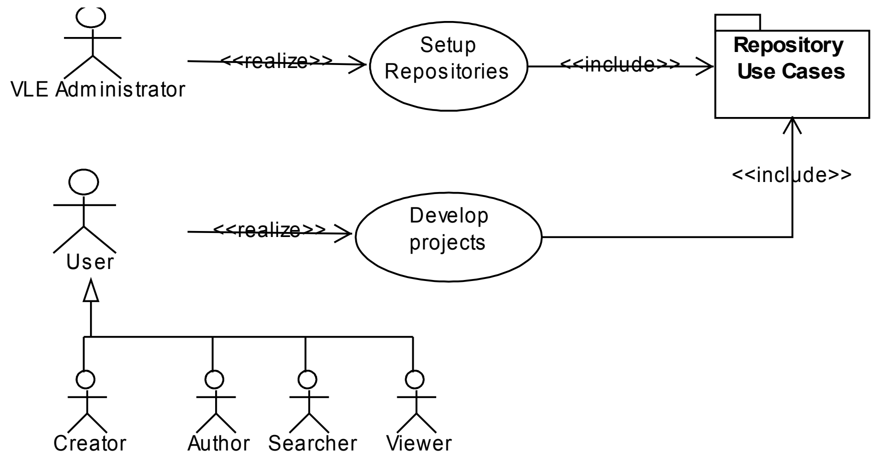

Fig. 16. General use cases view

Administrators must register the content repositories into the VLE registry. To perform this task, they must specify the URL where the web services interface as well as the web interface can be located. The web services interface consists of a WSDL document, in which the data types and the web methods are described. By the other hand, the web interface URL points to the repository start page. Once this information has been verified and the connection with the repository has been tested, users can access the services provided by the new repository. As mentioned on the architecture definition, users can have several roles on the system, depending on the tasks they are performing inside the VLE. Typically, users become Creators when they develop new contents, Authors when reusing previous contents and building more complex resources, and Searchers and Viewers while they are locating and visualizing the resources stored inside a repository. User actor encapsulates all these behaviors when developing project-based activities inside the Virtual Learning Environment (see Fig. 16).

4.2 Content Search/Reuse on Project-Based Work

In project based work, it is common to have students search and locate new resources, as well as develop documents to present the work they have done on the project . The sequence diagram Fig. 17 shows how project-based work tools on the VLE and content repositories can be used to increase the possibility of finding and reusing resources from several repositories.

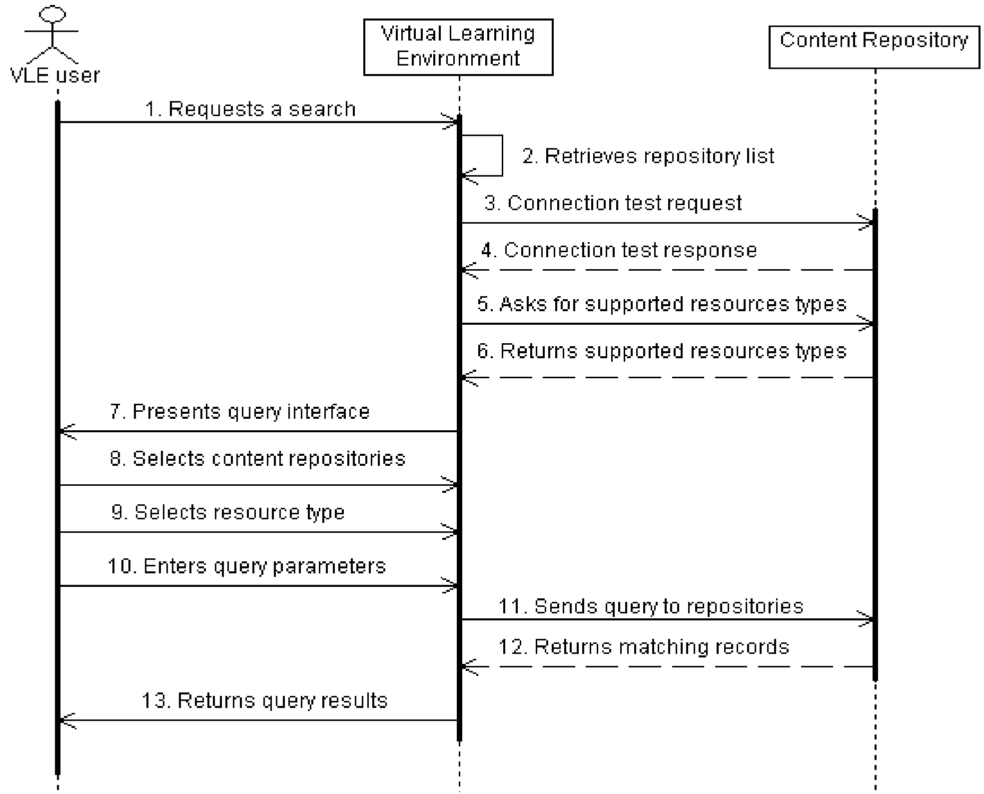

Fig. 17. General content search/reuse sequence diagram

The sequence diagram on Fig. 17 can be described as follows:

1. The project-based work tool allows users to add resources to a specific activity. So user requests for a query, in order to locate such resources.
2. The Virtual Learning Environment retrieves the list of repository in which the user is currently registered.
3. For each registered repository, the VLE invokes the connection test request method, in order to check if the repositories are available at the time.
4. Each repository answers the connection test request. Not accessible repositories will be skipped until a new search request is performed by the user.
5. For each active repository, VLE requests for the supported reference models and components. With this information it is possible to determine which resource types are valid on each repository.
6. Repositories return the supported resource types.
7. With the active repositories list and supported resource types for each repository, the VLE builds and displays the query interface to the user.
8. User must select the active repositories in which the query will be requested.
9. For each selected repository, the user defines which types of resources he/she is interested in.
10. Afterwards, user specifies the search pattern. This pattern can include special keywords, according to the capabilities of the repositories. Then, the user submits the query to the VLE.
11. The VLE receives the query, and it is forwarded to the selected repositories. To perform this task, the VLE automatically adds the user key for each repository.

12. Repositories return the matching records to the VLE. If no results were obtained, repositories return an empty result set.
13. After receiving the results from the repositories, the VLE builds the user interface and returns the results to the user.

5 Conclusions and Future Work

The design and implementation of a project based module made a significant improvement of current Aula Virtual services, by allowing teachers and students the improvement of complex learning activities that are motivating for both teachers and students. These learning activities will reinforce the current teaching and learning strategies that are being implemented as a support for traditional education on the Universidad Industrial de Santander, as well as other institutions that would like to use the VLE and the Content Repository.

However, project based work and content management are complex tasks, which require the creation of work teams that study and analyze their potential and the possible scenarios of application of these features. From this experience, a series of guidelines, manuals and tutorials must be created, in order to avoid the duplicity of efforts and to create a true learning community among several institutions.

The current implementation of the content repository offers the basic primitives for content object storage, management and search. Some improvements to be considered in near future involve security issues, supporting new reference models and specifications, and considering metadata schemas that are not represented as XML documents. Also, the integration of this solution with other local initiatives such as SPAR will be the first step towards the creation of the Edulatin Network.

References

[1] L. Machado and G. Ramos. Una propuesta metodológica de integración tecnológica al currículo ITIC. Universidad Pedagógica Nacional, Bogotá 2000.

[2] Grupo GENTE. Aula Virtual. http://tic.uis.edu.co/aula, April 2006.

[3] E. Meza. Aula Virtual 2.0, BETA Version. http://tic.uis.edu.co/aula2, April 2006.

[4] S. Tobón. Formación basada en competencias. ECOE editores, 2004.

[5] CONNECTING THE Bits: A Reference for Using Technology in Teaching and Learning in K-12 Schools. National Foundation for the Improvement of Education, Washington, DC 2000.

[6] J. R. Savery and T. M. Duffy. Problem Based Learning: An instructional model and its constructivist framework. Educational Technology, 1995.

[7] D. Ponta; G. Donzellini and H. Markkanen. Project Based Learning in Internet. In: 33rd ASEE/IEEE Frontiers in Education Conference. Boulder U.S.A., 2003.

[8] D. Moursund. ICT-Assisted Project-Based Learning. http://darkwing.uoregon.edu/~moursund/PBL. Last visited: March 20th, 2006.

[9] C. Castrillón. Prototipo para el Desarrollo de Programas de Especialización basados en Ambientes Virtuales de Aprendizaje. Master Thesis, Universidad Industrial de Santander, Bucaramanga, January 2006.

[10] Shareable Content Object Reference Model SCORM: Content Aggregation Model (CAM). Version 1.3.1. 2004. http://www.adlnet.org/downloads/70.cfm. Last visited 05-10-2005.

[11] IMS Digital Repositories Interoperability Specification Version 1.0. http://www.imsglobal.org/digitalrepositories. Last visited 05-10-2005.

[12] D. Rehak; D. Dodds and L. Lannom. A Model and Infrastructure for Federated Learning Content Repositories. In: Proceedings of Interoperability of Web-Based Educational Systems Workshop (WWW2005), 2005.

[13] E. Meza. Platform for the Exchange of Reusable Educational Content among Virtual Learning Environments. Master Thesis, Universidad Industrial de Santander. 2005.

[14] _______. Using Web Services to integrate Content Repositories and Virtual Learning Environments. In: Euro American Conference on Telematics and Information Systems EATIS, 2006.

[15] R. Heery and S. Anderson. Digital Repositories Review. UKOLN. 2005.

[16] Learning Technology Systems Architecture and Learning Object Metadata. Technology Standards Committee of the IEEE Computer Society http://ltsc.ieee.org. Last visited: 05-10-2005

[17] Dublin Core Metadata Initiative. http://dublincore.org/. Last visited: 05-10-2005.

[18] C. Cobos et Al. A Distributed Digital Content Repositories Model for Learning Objects in Latin-America. In: Euro American Conference on Telematics and Information Systems EATIS, 2006.

[19] PHP Hypertext Preprocessor. http://www.php.net. Last visited: 05-10-2005.

[20] JAVA Programming Language. http://java.sun.com. Last visited: 05-10-2005.

[21] MySQL Database. http://www.mysql.com. Last visited: 05-10-2005.

[22] E. Meza. Content Object Repository 1.0, BETA Version. http://tic.uis.edu.co/aula2/repository, April 2006.

[23] Google Web APIs Reference. http://www.google.com/apis/reference.html. Last visited: 16/11/05.

[24] SCORM Public-Access Repository. http://spar.unicauca.edu.co. Last visited: 27-05-2006.