Producing educational resources in the "libre" way: The Edukalibre project*

Jesus M. Gonzalez-Barahona
Grupo de Sistemas y Comunicaciones
Universidad Rey Juan Carlos
Mostoles, Spain
jgb@gsyc.escet.urjc.es

Vania Dimitrova
University of Leeds
School of Computing
Leeds, UK
vania@comp.leeds.ac.uk

Diego Chaparro
Grupo de Sistemas y Comunicaciones
Universidad Rey Juan Carlos
Mostoles, Spain
dchaparro@gsyc.escet.urjc.es

Chris Tebb
School of Computing
University of Leeds
Leeds, UK
chrispy@comp.leeds.ac.uk

Riccardo Mazza
Università della Svizzera italiana
Lugano, Switzerland
riccardo.mazza@lu.unisi.ch

ABSTRACT

Recently emerging methodologies for producing educational resources resembling those used in the libre (free, open source) software will radically change the way educational content is developed and used on the web. To fully implement the libre idea, both educational practitioners and students should become actively involved in the creation and distribution of open resources. New architectures are needed to effectively support this process. The paper describes a novel, *truly open* platform to support the creation of free, collaboratively constructed educational content on the web. The platform has been developed within the Edukalibre project. The Edukalibre system provides easy access to core technologies composed of a control version system combined with conversion tools to produce several convenient formats for each document. Its modular architecture offers many different interfaces to the users. The Edukalibre platform is distributed as libre software.

Categories and Subject Descriptors

K.3.1 [**Computers And Education**]: Computer Uses In Education—*Computer-managed instruction*

General Terms

Design, Human Factors

Keywords

Architectures for web-based education delivery environments, Collaborative development of learning material, Libre software development model, Open source, Control version systems

1. INTRODUCTION

Most educational institutions, especially universities, are exploring the possibilities of applying web-based information systems to education. Currently, such systems are used for re-enforcing or complementing traditional teaching techniques, but, in many cases, new educational paradigms are emerging. New methodologies and architectures are needed to provide effective ways for the creation, sharing, and re-use of educational resources on the web, which can be deployed in a variety of educational paradigms.

This paper discusses recently emerging methodologies for producing educational resources resembling those used in the libre (free, open source) software[1] projects. The libre software development method has tremendously changed the way software is being produced and deployed [1, 10]. Strong communities of practitioners who share experiences, methods, code, and constantly help each other have been built in a number of domains, ranging from operating systems (Debian, FreeBSD, Fedora) to desktop environments (GNOME, KDE), web browsers (Mozilla, Firefox), web servers (Apache) or office suites (OpenOffice.org) [6, 9, 22, 15]. It is now clearly recognized that the libre software methodologies have led to revolutionary methods of producing programs and advancements in software development [21, 13].

Similarly, the libre methodolgy may lead to innovative educational paradigms and will have a great impact on the way the web is used for teaching and learning. Thus, it

*The work described in this paper was financed in part by the European Commission's, Socrates/Minerva program, under grant number 110330-CP-1-2003-1ES-MINERVA-M. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

WWW2005, May 10–14, 2005, Chiba, Japan.
.

[1]in this paper, we will use the term "libre software" to refer both to "free software" and "open source software". The term "libre" lacks the ambiguity of "free" (which means both "gratis" and "free as in freedom"), and makes reference to the liberty of change that the user of the software gains with it. In short, libre (free, open source) software is defined by the freedom of use, study, modification and distribution that is granted to those receiving a copy of the programs.

is extremely important that the educational community becomes aware of the potential of the libre development model. There is a strong demand for studies that examine ways of applying the libre software model in educational settings, as well as for technologies that support the effective deployment of this model. Examining the connection between libre software development and creation of open content for education is the main objective of the work presented in this paper. We will describe a study being conducted within a EU funded project that involves developers and educationalists from several European countries working together to study both technological and pedagogical aspects of the successful deployment of the libre idea in university teaching.

There is a growing interest among the educational community to adopt aspects of the open development model. The first step is being undertaken by leading educational institutions, such as MIT [19], Carnegie Melon [11], and Harvard [20]), which challenged the traditional way of thinking, (according to which teaching materials were available only to students enrolled in the courses) and provided freely available, high-quality academic content on the web. Economical and technological factors favor this approach. Internationally available web-based services can change the landscape of tomorrow's education. Material can be made available to students all over the world at no (significant) additional cost, with many benefits for the institution providing the material. The *open learning content* idea is being taken on board with great enthusiasm [24], and implemented in a number of projects, e.g Open Learning Support [25], MIT OpenCourseware [19], Open Learning Initiative [11], Connextions [12], CASCADE [23], WIKI [29]. Furthermore, there is a growing interest in methods for sharing open source content outside the academic communities, for example schools [28] and digital libraries [3].

To fully develop the libre software idea, both educational practitioners and students should become actively involved in the creation and distribution of open content. Until recently, technical challenges made it very difficult to support truly open, dynamic, educational resources constructed collaboratively by large groups of teachers and even students. Nowadays, the libre software community has created a vast amount of technologies to support their practices. However, these technologies have not been explored fully in an educational context. Moreover, many of the existing technologies for collaborative development of software suffer from two problems: (1) they are suitable for software developers but are not intuitive enough to be adopted by teachers and students and (2) they normally tackle small tasks, which is convenient for developers who usually use several tools, but is inappropriate for teaching and learning communities.

New architectures are needed that effectively support the collaborative construction of open educational resources on the web [18]. Moreover, these architectures should themselves be open, enabling customization and deployment in different settings. An example of such an architecture is proposed in this paper. We describe a novel, *truly open* platform to support the creation of free, collaboratively constructed educational content on the web, which has been developed within the Edukalibre project. The project is aimed at examining the connection between libre software development and creation of open content for education. It is funded by the European Commission under the Socrates/Minerva program[2], started on October 2003 and is expected to last until December 2005. The project is coordinated by University Rey Juan Carlos (Spain), and includes as partners teams from University of Leeds (United Kingdom), University of Porto (Portugal), University of Karlsruhe. Project web site: http://www.edukalibre.org

The paper will first discuss the link between libre software development model and the production of educational materials (Section 2). We will then present, in Section 3, the architecture of the Edukalibre system. Some specific applications of the system will be shown in Section 4. Section 5 will describe the current deployment of the system in university teaching. Finally, in the conclusions, we will sketch some plans for future work.

2. LIBRE SOFTWARE DEVELOPMENT AND EDUCATION

Since early 1980s, the libre software community has shown how software systems can be produced by tightly linked groups of people collaborating over the Internet. Participants in this community have developed over the years a set of practices, procedures and uses for building programs. This set is usually referred as "the libre software model". The Edukalibre project explores how this model can be translated to the education camp, and specially to the collaborative development of educational materials. Therefore, we will first outline the key characteristics of this libre software model.

To begin with, it is important to notice that there are nearly as many libre software development models as libre software projects. Variety is one of the most relevant characteristics of the libre software community. However, there are several specific issues that are found across most libre software projects:

- Frequent and early release of the software, even when it is not fully tested. Eric S. Raymond spells this out as "Release early, release often" [21]. The rationale is that when the work is early and frequently shared with the community, there are more opportunities to gain external feedback and contributions.
- Quality by inspection of many individuals, including those outside the group of developers. Raymond (also in [21]) names it the Linus Law: "Given enough eyeballs, all bugs are shallow". That is, defects can be found by many people, and if appropriate mechanisms and chances are provided, that "peer-review" can detect many problems that would otherwise go unnoticed.
- Development is done by geographically distributed groups. It has been shown several times (see for instance [7]) how libre software projects, especially when of a certain size, are carried out by a number of geographically distributed developers. They seldom (if ever) see each other face-to-face, and work in a coordinated fashion without formal hierarchies.
- Development is done usually asynchronously and only using software tools to coordinate [6]. Those tools

[2]`http://europa.eu.int/comm/education/programmes/socrates/minerva/ind1a_en.html`

(mailing lists, bug tracking systems, version control systems, software repositories) are commonly deployed in websites offering hosting facilities to libre software (such as SourceForge[3] or Savannah[4]). All the coordination and communication is usually logged or archived for future use, and is publicly available over the Internet.

- There is a mixture of voluntary and paid work, particularly in large projects, and in projects of interest for certain companies.

The Edukalibre project is exploring how this model can be applied to education, and specifically to the creation and maintenance of educational materials. This means that we explore a model resembling the one of libre software development, but with the particular characteristics of the educational (and more specifically the University) context:

- Educational materials will be located mainly on the Internet, in the context of an LMS (Learning Management System). This is hardly a novel issue, but gives the needed context for the rest of the model.

- Materials will be produced by groups of educators, usually in different institutions, and geographically dispersed. Since the curricula is similar in many cases, it seems reasonable that different teachers have similar needs, and can collaborate in the making of their materials. That collaboration has the implication of sharing the work, but also of coordination and consensus about decisions. Similar situations are found in libre software projects, and it is expected that similar results can be obtained. In the current European context, where collaboration among Universities of different countries is encouraged at many levels, this will become more and more a common situation.

- Materials will also be used, commented, and maybe modified by students. In fact, in many cases, students are already producing their own materials, based on those provided by educators. In the same way that users can contribute to libre software projects highlighting bugs or even proposing useful modifications to the programs (if they have enough knowledge), students (users of the materials) can contribute a lot to improve the quality in a continuous process. From this point of view, frequent and early release could be as important as it is in the case of libre software.

- Educators and students will need simple to use, yet powerful, tools to be able to collaborate in the way libre software developers do. For instance, it will be important to mimic the functionality of version control systems and compilation systems, specifically oriented to the production of educational materials. Authors should be able to use tools common to them, but in such a way that coordination and work in common becomes an intuitive process. Authors of educational materials are not (usually) programmers, and their familiarity with software tools is by no means of the same order of magnitude.

[3]`http://sourceforge.net`
[4]`http://savannah.gnu.org`

- The public availability of produced materials will enable the collaboration of third parties, such as other professors or students from other institutions that could find the materials useful, and worth contributing to.

- Several licensing terms will be explored, some of them allowing for publishers to take the materials and distribute them for profit, in the same way that there are companies distributing GNU/Linux based systems composed only of libre software. The current interest created by some initiatives related to open content (such as Creative Commons[5], the Open Archives Initiative[6], and the the Public Library of Science[7]) is caused by many other projects exploring, particularly in the scientific context, the advantages of this approach.

Of course, this model is difficult to explore without the proper tools and some user groups ready to test and try it. In the following sections it will be shown how the Edukalibre project is building such tools, and how some test experiences have already started, which are being studied and evaluated within the project.

3. THE EDUKALIBRE SYSTEM

The Edukalibre project has already released the first version of a system that allows for the flexible management (via the web) (using HTTP and WebDAV protocols) of a set of documents. The system is designed to let users choose from a wide set of tools (from easy-to-use word processors, such as OpenOffice.org[8], to less common XML-based editors), and includes version control facilities and automatic conversion to many end-user formats (ranging from PDF, ready to be printed on paper, to decorated HTML, suitable for previewing when creating a new version or for reading online).

A diagram of the architecture of the whole system is presented in Figure 1. It shows how the Edukalibre platform is organized around a *document repository*, which is used for supporting collaborative editing, and also provides the basis for a web-based *groupware tool for collaborative authors*. In addition, some extensions are provided, such as a web based *activity monitoring tool* for teachers. Modules external to the repository can access it directly, using HTTP and WebDAV protocols, or by means of a well defined API. This is the basis of the modular nature of Edukalibre: adding new modules which provide new interfaces to users is a simple process for developers.

The document repository is built on top of a *version controlled storage system*, with document conversion and management tools written in `Python`[9] [27]. The system has been developed to be as open as possible, using libre software technologies throughout and allowing access to content in as many different ways as is practical, respecting the many and various platforms and working methodologies that collaborative authors use today.

From the perspective of an author collaborating to write some material, the platform provides a simple interface based

[5]`http://creativecommons.org`
[6]`http://www.openarchives.org/`
[7]`http://plos.org`
[8]`http://openoffice.org`
[9]`http://python.org`

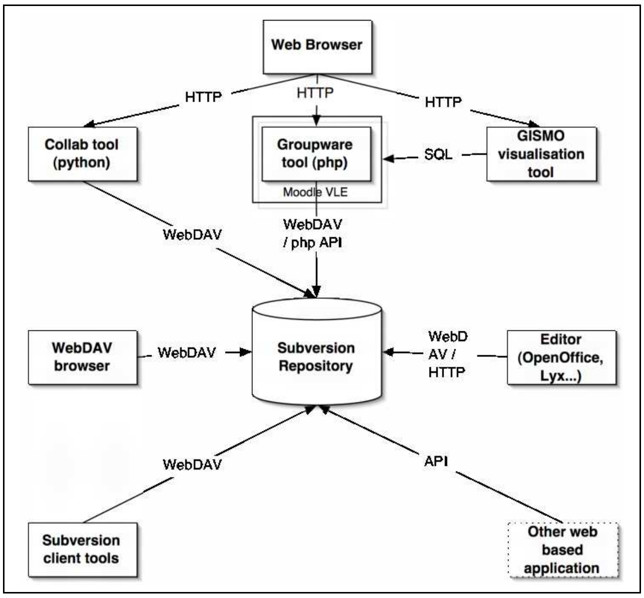

Figure 1: Architecture of the system

on a web browser (although other interfaces are also provided, including some integrated with an LMS). From the perspective of the systems developer wishing to incorporate access to an Edukalibre repository into their applications, the platform provides a simple PHP[10] API for web development, and Subversion/WebDAV access to the repository for other development platforms.

All the code used and produced by the Edukalibre project is libre software. This permits the easy replication of the system anywhere, with or without the knowledge of the project. Some other groups not in the project have already expressed their interest in installing their own versions to start their own experiences, which is welcome and promoted by Edukalibre.

With respect to the design principles, the architecture of the system is intended to be simple, yet flexible, reusing as many already available components as possible, so that the project can focus on its goals whilst incorporating already tested and widespread components, such as the Subversion[11] version control system or the Moodle[12] LMS. But those components (and others also used) could be easily exchanged for others of similar functionality. In fact, where possible, the project is already providing more than one choice to implement each component, as will be shown later. And of course, the use of standard protocols for the communication and access to the different components of the system has also been a core design requirement.

3.1 Review of technologies

The system is composed of many different software components, most of them widely used in current libre software projects, ensuring their stability and maturity. When looking at the platform as a whole, we find some of those external modules (libre software code not developed by Edukalibre), some components developed specifically by the project, and some middleware code to link everything together.

The main software components within the Edukalibre project are:

- *Control version system*: Every document created in the system has a main version and a history of older revisions or branches. Each revision can be accessed or modified at any time, and any old revision can be set up as the main version of the document, or assigned as a new branch (for example to adapt higher education teaching materials for a secondary education audience).

  All of these processes can be performed by the control version system used. It allows us to keep different versions of a document, but not only that, it also allows to realize some other goals that are hard to satisfy without it, such as the integration of different versions of the same document.

  Currently, the platform uses Subversion as the control version system. Subversion is in some sense an evolution of the more traditional CVS[13], which fixes many of its problems, and adds new features.

- *Editors and word processors*: The main use of the system is to create documents, therefore a very important component of the system are the editors used to create these documents. Our goal is not to create new editors, but to provide ways in which existing editors and word processors can interact in the most flexible way with the system, and to provide the flexibility to allow a greater variety of editors to be integrated.

  OpenOffice[14] is an office suite which includes an WYSIWYG word processor. It can edit, in addition to files in its own format, DocBook/XML[15] files (with the help of some code which the project has improved) and Microsoft Word documents. It is widely popular in software libre community, and its use is growing quickly everywhere. The current Edukalibre platform allows a document to be opened, modified and stored using just OpenOffice, with the user not needing any other tool.

  However, any other word processor can be used to modify the documents of the system, if it can manage DocBook/XML or LaTeX[16], the currently supported document formats. Since both are plain-text formats, any text editor can also be used, provided the user knows the syntax of the format. A simple web editor is also provided in the system to facilitate the modification of documents using exclusively a web browser, and a wiki-like system supporting conversion from wiki format to DocBook/XML and vice-versa, is now in development.

- *Tools for format conversion*: Some other important pieces of software are those in charge of converting documents from the base formats to others, more suitable for reading (such as HTML or PDF). Most of

[10] http://php.net
[11] http://subversion.tigris.org/
[12] http://moodle.org
[13] http://www.cvshome.org
[14] http://www.openoffice.org/
[15] http://www.docbook.org/
[16] http://www.latex-project.org/

the conversions from XML formats are performed using XSLT[17] stylesheets. The XSLT processors used are xsltproc and xalan[18]. Most of the needed XSLT stylesheets were already written, but more can be added, and our existing stylesheets are being rewritten or improved to better meet our objectives, such as those to convert between DocBook/XML and OpenOffice format. Conversions to PDF and Postscript are done using TeX/LaTeX and Ghostscript[19].

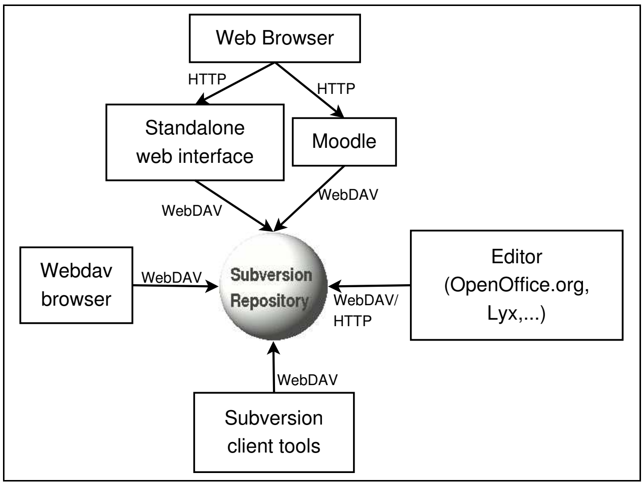

Figure 2: Components of the collaborative editing subsystem

- *Web interfaces*: One of the goals of the project is to allow for as much different interfaces to the system as possible, and to let users decide which are their preferred writing tools. In addition to accessing the system directly with a text editor plus a Subversion client, other interfaces already provided are:
  - A collaborative editing system (COLLAB), which is a simple Python-written web interface that permits to perform common actions on the system, such as listing of documents or visualization of the history of a document.
  - The same collaborative editing system has also been integrated as as a component of the libre software Moodle VLE, providing all the additional functionality of a learning environment to support the collaborative editing process.
  - There is also a PHP groupware application written as a component of Moodle, called ConDOR. Its aim is to examine the effectiveness of the tools within a simple groupware application, also showing the flexibility of the system to allow developers from different communities and with different backgrounds (PHP, Python, WebDAV) to integrate the Edukalibre systems with their applications.

  Both COLLAB and ConDOR will be described later in more detail.

- *Communication protocols*: Standard protocols are used to connect the different components. They are basically the ubiquitous HTTP [4], and WebDAV [8] (an extension to HTTP tailored to the needs of collaborative editing and management of files on remote web servers).

Figure 2 shows the different components of the collaborative editing subsystem and their main relationships.

The Edukalibre system offers a PHP API to allow the easy integration of any PHP application with it. It uses the libre software VersionControlSVN package, which is a part of the PEAR [20] PHP libraries. PEAR is a core PHP group project with the goal of producing reusable APIs for rapid application development, similar to the Standard Template Library for C++. VersionControlSVN provides a simple way to access the contents of a Subversion repository, and returns its data in simple arrays, familiar to web programmers.

The Edukalibre PHP API builds on this to provide simple XHTML form generation for interacting with the Python scripts that control the repository. It also provides a comprehensive set of file management tools for folder creation, file deletion, history extraction and functions that return the differences between certain versions of a given document. A sample PHP web application is provided, that allows for full access to the functions of the repository, composed of just 70 lines of code.

3.2 Formats

The central elements of the Edukalibre system are the documents. Every document can be accessed in many different formats, but it is convenient to distinguish between two kinds of those formats:

- Base formats, which are those that can be the inputs for the whole system. A document can be produced or modified in any of these base formats. At the moment the base formats are: DocBook/XML, OpenOffice.org and LaTeX. They have been selected for their properties and advantages. All of them are plain text formats, which is important for easy management under version control systems. They also allow for methods to automatically extract information (such as the title or the authors) form the document, and for all of them exist mature libre software tools to convert them to end-user formats.

- End-user formats, which are automatically generated by the system from the base formats. These include PDF or Postscript (both suitable for printing), HTML in several flavors (in a single page, in multiples pages, or even decorated with the desired web page style) or plain text. Some base formats are generated from other base formats.

It is important to notice that only base formats of the documents are stored in the control version repository. The end-user formats are stored independently, in the end-user repository, since they can be generated from the corresponding base format. The schema for conversions between formats is given in Figure 3.

[17]`http://www.w3.org/TR/xslt`
[18]`http://xml.apache.org/xalan-j/`
[19]`http://www.cs.wisc.edu/~ghost/`
[20]`http://pear.php.net/`

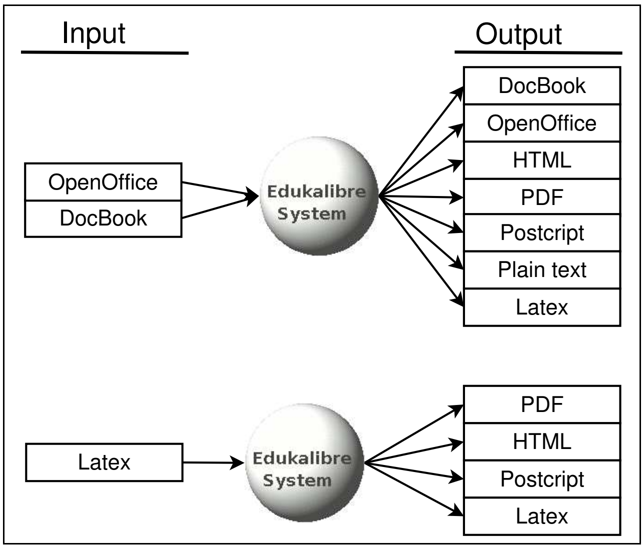

Figure 3: Document formats

3.3 Repository manager

The repository manager is the core of the Edukalibre platform, and also the heart of the collaborative editing tool (see section 4.1). It provides the main functionality of the system, being in charge of storing the documents, creating the end-user formats, and providing information about the documents stored in the system. It can be split in two parts: the *core repository*, and the *document conversion and metadata extraction system*.

- The core repository is basically a storage system in which documents, together with information about them, are stored. This repository consists basically of two elements: the version controlled document repository, in which the base formats are stored, and the end-user repository, in which only the end-user formats are stored in an HTTP server. The document repository is a subversion repository, with aditional plugins to perform some core tasks when a document is uploaded. It stores only the base formats. The second element is just an HTTP server, which stores the end-user formats of all versions for all documents.

- The document conversion and metadata extraction system provides the functionality of the repository manager. The normal process operation of this system is launched when a new document or a new version of an existing document arrives to the repository manager via any of the different interfaces shown in Figure 1. From there on, it follows the following steps. First of all, the system accepts only valid files. A validation has to be done on the document to check if it can be converted correctly to end-user formats. If the document is valid, it is stored in the subversion repository, and is converted to the end-user formats, which are later stored in the end-user repository. At the end of the process, some information is extracted from the document, and also stored in the system (such as the title or the authors).

This system also provides methods used by the different interfaces to extract metadata information about the content of the repository manager: listing of documents stored in the system, information about history of each document (versions, authors, dates, etc), logs about the conversion of each end-format for each document and general properties about each document (base format, title, abstract, etc).

4. APPLICATIONS

In the following subsections, some applications that are part of the Edukalibre system are described. All of them show both how the functionality of the system can be integrated with external tools, and also how the architecture is modular enough to make room for may different ways of interaction.

4.1 COLLAB: a Python-driven Web Interface

This is one of the possible interfaces to the system, and allows the users to perform some of the common tasks using just a web browser. A screenshot of it can be seen in Figure 4.

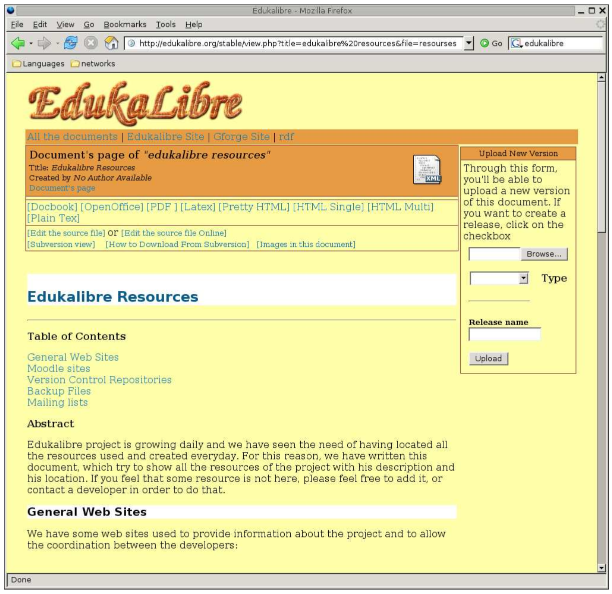

Figure 4: Collaborative editing tool interface

The main page of the interface consists in a list of documents created in the system, with some information about each of them. The interface provides a way to show complete information about each document, and to create new documents. An RDF channel with information about the documents recently modified or created is also provided. The information for each document includes:

- Listing of its version history, with the release date and the comments from the release author.

- Listing of links to all end-user formats for every version, so that users can get any format of any version of every document.

- Editing and downloading options for each version, including the possibility to download the base format, modify it and upload the new version using the web interface, or to edit it on-line with a simple web editor. Some information about how to modify and update the document using some other interfaces, such as a web editor with WebDAV capabilities, or through the subversion client tools, is also provided.

- Forms to update the document, uploading a new version (written with a standalone editor such as OpenOffice.org).
- Decorated HTML version of the latest version of the document, to show a quick preview of it, with all the basic functionality to manage it.

The interface also provides a form to upload a new document to the system.

4.2 ConDOR: An Intuitive PHP-driven Groupware Application

One of the core deliverables of the project was to create a *user friendly* working environment which would allow us to evaluate the effectiveness of web-based version controlled collaborative authoring, see Figure 5.

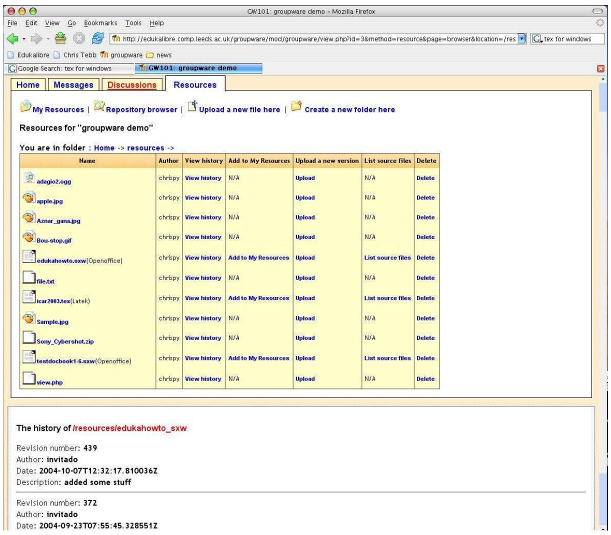

Figure 5: Groupware tool for constructing of dynamic open resources (ConDOR).

To this end, a bespoke groupware application was produced, which ran as a modular component of the libre software learning management environment Moodle. The groupware was written in PHP, and made use of the PHP subversion API which we created to allow easy access to the document repository for PHP web application developers.

The requirements specification for the groupware was taken from a questionnaire that was filled in by all partners of the project, enquiring into the scenarios in which the platform would be tested.

The majority of the official evaluation scenarios were based around a teacher assigning collaborative group coursework for students, and the students using the groupware tool to produce shared resources, take part in discussions about the work and produce a single final document for submission. The basic requirements of the users were:

- Access to shared resources
- Access to discussion facilities
- Easy version control

The groupware application allows easy "explorer style" access to the document repository, allowing intuitive navigation of the file and folder structures, and simple tools for uploading files and creating new folders.

With the current prototype, each groupware is linked to a fixed repository, but the system will allow the users to browse the content in multiple arbitrary repositories. To prevent excessive repeated navigation of large folder structures, a "My Resources" section is provided within the groupware resource area to allow authors to bookmark documents deep within the resource tree that they have a current active interest in.

From either the main resources browser or the "My Resources" bookmark area, authors can download a document in any supported format (currently XHTML, PDF and plain text), upload a new version of the document, browse the history of the document and add/remove documents from their "My Resources" list.

The repository will allow authors to add any type of file, and put it under version control. Extra features such as format conversion are only available for OpenOffice and LaTeX documents, but trial users have been using the repository to manage many kinds of file. A graphic designer using the system used it to manage multiple versions of an Adobe Photoshop document he was working on with another designer and a systems administrator has used the system to manage various versions of Windows configuration files.

Current work on the groupware application includes implementing online editing of XHTML files (with a Java applet), and wiki style editing of OpenOffice documents (via a DocBook/XML conversion utility).

It is the aim of the developers to provide as unrestrictive a set of authoring tools as possible. Currently, the system supports offline authoring of any file type by allowing users to download a file, work on it using a local application and upload a new version. The next phase of development will see online editing and simple visual change management (similar to the programmers tool diff) of XHTML, LaTeX, OpenOffice, DocBook/XML and Microsoft Word documents.

4.3 GISMO: a Graphical Interactive Students MOnitoring tool

As shown in Figure 1, the Edukalibre system is currently linked to the Moodle Learning Management System. Educational materials collaboratively produced by using COLLAB or ConDOR are made available to the students through Moodle. This is a convenient way to immediately deploy educational materials to the students without the hassle of exporting the content in some format and upload it to a server on the Internet. Instructors may use these materials on their courses given on-line with this platform.

Student tracking data provided by the Learning Management System is a valuable source of data that can be used by the instructor not only to check students activities, but also to improve quality of the materials. For instance, an instructor may check which part of the course's materials are most and less accessed by the student, and then perform further investigations to understand whether the students found these parts difficult to understand or not. Furthermore, the students activities in ConDOR are recorded and can be used to analyze the collaborative work of a group of students or teachers who are constructing resources using the Edukalibre system.

Student tracking data is complex and is usually organized in some form of a tabular format, which is in most of the cases difficult to follow and inappropriate for the instructors' needs [17]. For this reason, a graphical and interactive tracking tool, GISMO, was implemented as an application that interfaces with the Edukalibre system. GISMO uses the students' tracking data as source data, and generates graphical representations that can be explored and manipulated by course instructors to examine social, cognitive, and behavioral aspects of distance students. It implements some of the visualizations designed in our earlier work and found useful by teachers [17] within a new context, namely the Edukalibre system.

Gismo processes students data and generates graphical representations thought a pipeline of several stages:

- The *Data Exporter* collects any possible data regarding the activities done by students (tracking data) and exports this data into a MySQL database (*the GISMO repository*). This module can be adapted to other Learning Management Systems, such as Claroline, Fle3, Mimerdesk, etc.

- To represent data in visual format we need to do some computations and transform the data into visual structures and representations. This step is performed in GISMO by the *Data Processing and Visualisation Procedure*. This "visualisation pipeline", follows the well known "reference model for visualisation" proposed by Card et al.[2]. The modules in the visualisation pipeline are implemented in Java. In particular, views are implemented and delivered through the Web using a Java Applet.

- Instructors can investigate the users' reading of course materials, which can give invaluable feedback on the quality of collaboratively constructed resources and their practical use by students. Figure 6 shows an example of GISMO visualization that can be used by instructors to get insights of how popular is the content material to the students of the course. It shows, for a specific part of the course content, in which days the students accessed this material of the course and how many times. The chart on the top illustrates accesses for each student. The histogram on the bottom represents the sum of accesses made by students to the resource on each specific date.

5. DEPLOYMENT OF THE EDUKALIBRE SYSTEM IN UNIVERSITY TEACHING

The Edukalibre project is scheduled in two main phases. The first one is devoted to the building of a proof-of-concept system which can be used to explore the new development models of educational materials. At the moment of writing of this paper, the project is in the transition from the first to the second phase. A deployable version of the system described in the previous sections has already been completed and is being deployed in real settings.

We are now conducting formative evaluation [16] to examine the system's functionality and usability. It includes both *expert inspection* and *user trails* of the three applications described above, namely COLLAB, ConDOR, and GISMO.

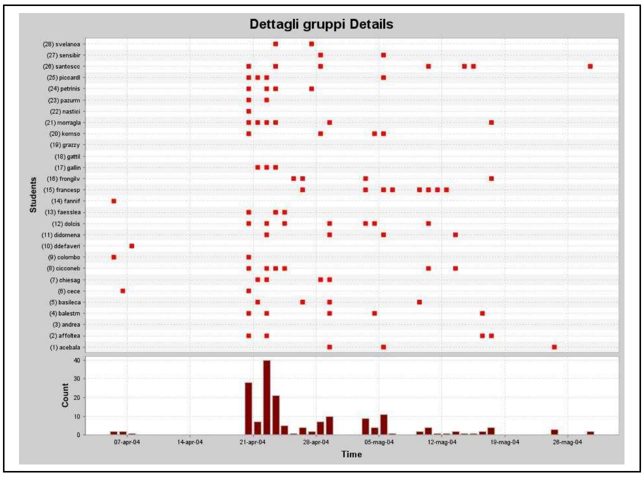

Figure 6: Representation of the students accesses to a particular content material of the course

The *expert inspection* is based on a framework for usability evaluation of e-learning web applications, called MILE [26], developed by a team of Human-Computer Interaction researchers from the Politechnico di Milano and the University of Lugano. The inspection is driven by typical educational scenarios, which were defined based on a questionnaire involving university teachers from all Edukalibre partner institutions. The scenarios, together with possible user profiles, were used to determine inspection criteria. Twelve experts, including teachers and human-computer interaction specialists, have been asked to inspect the Edukalibre applications, and to fill in MILE usability sheets. This evaluation is on-going.

Formative evaluation can be addressed properly only if inspection is combined with real user trials. In line with the Edukalibre objectives, the initial user trials consider the creation of open educational content by both user groups - teachers and students. During the last two months the Edukalibre applications are being used within partner institutions in the following situations: several authors working together to prepare educational content, researchers creating a repository of papers and collaboratively written reports, and students creating repositories on a common topic of their dissertation projects.

Some preliminary comments from inspectors and users have already highlighted issues that are being fixed by the Edukalibre development team, such as:

- *flexibility* - which refers to the robustness of the system to deal with a variety of file formats, at the moment, version control is being fully enabled for OpenOffice and Latex documents;

- *effectiveness* - which refers to the applicability of the system in learning scenarios, and required several iterations to ensure that applications complied with the tasks they were intended for, i.e. collaborative writing by teams of teachers or students (COLLAB and ConDOR), resource management for group projects (ConDOR), and monitoring of student activities (GISMO);

- *efficiency* - which refers to how quickly can the users perform their tasks and resulted in fixing some cumbersome operations and long navigation menus;
- *user satisfaction* - which refers to creating subjectively pleasing systems, and required the fixing of some design and aesthetic issues.

During the second Edukalibre phase, a summative evaluation of the improved system will be conducted in real settings at partner institutions. These include:

- Students working in groups to take a decision on an environmental case and write a group report to justify their decision - this is being prepared by the partners from Karlsruhe;
- Students researching and producing content on a new topic, including collaboratively written review and a collection of resources on the web - this is being prepared as part of a module on Personalization and user-adaptive systems at Leeds;
- Educationalists from geographically remote locations constructing collaboratively educational content on statistical methods and operational research, which includes teachers and researchers from Madrid, Prague, and Porto.

The above situations will utilize COLLAB and ConDOR for collaborative writing and group work. In addition, the student's performance and work with the educational resources will be monitored with GISMO.

6. CONCLUSIONS AND FURTHER WORK

The Edukalibre project is exploring the field of collaborative production of educational materials from a novel point of view: to which extent the common practices of the libre software community can be translated to the educational content domain. For this exploration, the project has developed (mainly by reusing already existing components) a web-based system which aims to provide some of the functionality available in version control systems, but specifically targeted to the producers of educational documents. With this target population in mind, an effort has been done to minimize the learning effort for using the tools, yet maintain as much functionality and flexibility as possible.

The system is built around a Subversion-based repository for document sources (written in DocBook/XML or LaTeX), together with access methods to it (using standard protocols such as WebDAV and HTTP) and code to automate the generation of final formats (such as HTML or PDF). In addition, translation to and from OpenOffice.org format is provided, which (in addition to the WebDAV access) allows users to interact with the system from within an office suite, without having to deal with the complexities of a regular version control system plus conversion utilities.

The Edukalibre architecture is modular in nature, and makes it simple to add new functionality by connecting new subsystems. Some of them are currently provided: COLLAB and ConDOR (designed to provide extra tools for collaboration among writers of a document) and GISMO (for the monitoring of student activities). In addition, the system can also be used in the context of an LMS tool, and, indeed, the integration with a popular open source LMS, Moodle, has already been included.

The system as a whole allows for the easy modification of the documents by educators, or even by students (if the have the required permissions). New versions of the modified document are automatically generated, and immediately put at the disposal of readers. In some sense, the system implements a wiki-like syntax, with simple to use interfaces integrated in environments (such as office suites or LMS systems) common to educators. All in all, it lowers the technical barriers for educators to explore the libre software development model applied to educational documents. Also, the whole system is composed of libre software, which makes it easy to try, install and adapt to any environment.

There are other projects which deal with some of the aspects important for Edukalibre. Among them, it is worth mentioning Apache Forrest[21], which is a documentation framework which can be used to render source content in several formats (including a subset of DocBook/XML) into a complete, customized website. Some ideas of the architecture for converting and managing formats are taken from it. It is also relevant to consider BSCW [5], a cooperative application which supports cooperation through small repositories in which users can upload documents, hold discussions and coordinate their work with other user. However, no version control in the sense provided by Edukalibre, or complex conversions to several formats is provided. In some sense, BSCW could be the basis for some of the tools that Edukalibre misses to replicate some of the functionality provided by GForge (such as discussions or coordination tools). Another system with some resemblances is Connexions [12], which offers a collection of free educational materials and a set of free software tools to help authors, instructors and learners (but is more a collection than a supporting system, like Edukalibre). Another example is the SERUM [14] system, which proposes a model of customized repositories for collaborative authoring of web content, and in this is similar to the Edukalibre system presented here.

Currently, the Edukalibre system is entering its second year, when in addition to some evaluation activities, new lines of work are to be open. Among them, some of the most relevant are the exploration of more complete version control patterns (including the use of private branches, for instance for groups of students maintaining their own customized and commented version of a document); the improvement of the different interfaces and the integration with OpenOffice.org; the improvement of the integration with LMS systems such as Moodle; and the integration with wiki-like tools (considering wiki formats as one more format to convert to/from the source documents).

In the future, more work can be done in the direction of integrating more and more subsystems common in libre software development (and in software such as GForge[22]), such as bug tracking systems (for unresolved problems with document content), activity trackers, release management systems, etc. In the end, it would be interesting to consider the construction of hosting sites for the creation and management of documents and other types of materials, in the same sense that sites like SourceForge[23] provide support

[21] `http://forrest.apache.org/`
[22] `http://gforge.org/`
[23] `http://sourceforge.net`

and hosting for libre software projects.

Finally, it is worth noting that this paper was written collaboratively with all authors being actively involved throughout the whole writing process. It gave us the opportunity to truly appreciate the potential of the libre development idea and to experience its application in a practical task, the result of which is this paper. This small practical experience gave us a subjective confirmation that systems like Edukalibre are indeed very much needed in the creation of open educational content.

7. REFERENCES

[1] Nikolai Bezroukov. Open source software development as a special type of academic research. *First Monday*, 4(10), October 1999.

[2] K. S. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization, using vision to think*. Morgan Kaufmann, California, USA, 1999.

[3] Reid Cushman. Open educational content for public digital libraries. Technical report, The William and Flora Hewlett Foundation, 2002.

[4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – HTTP/1.1, RFC 2616. Technical report, Network Working Group, IETF, June 1999.

[5] BSCW: Basic Support for Cooperative Work. `http://bscw.fit.fraunhofer.de/`.

[6] Daniel M. German. The GNOME project: a case study of open source, global software development. *Software Process Improvement and Practice*, pages 201–215, August 2003.

[7] Rishab Aiyer Ghosh, Gregorio Robles, and Ruediger Glott. Software source code survey (free/libre and open source software: Survey and study). Technical report, International Institute of Infonomics. University of Maastricht, The Netherlands, June 2002. `http://www.infonomics.nl/FLOSS/report`.

[8] Y. Goland, E. Whitehead, A. Faizi, S.R. Carter, and D. Jensen. HTTP extensions for distributed authoring – WEBDAV, RFC 2518. Technical report, Network Working Group, IETF, February 1999.

[9] Jesús M. González-Barahona, Luis López, and Gregorio Robles. Community structure of modules in the apache project. In *Proceedings of the 4th Workshop on Open Source Software Engineering. 26th International Conference on Software Engineering*, Edinburgh, Scotland, UK, May 2004.

[10] Ahmed E. Hassan, Michael W. Godfrey, and Richard C. Holt. Software engineering research in the bazaar. In *Proceedings of the 2nd Workshop on Open Source Software Engineering at the 24th International Conference on Software Engineering*, May 2001.

[11] Carnegie Mellon Open Learning Initiative. `http://www.cmu.edu/oli/`.

[12] Connextions: Sharing knowledge and building communities. `http://cnx.rice.edu/`.

[13] Stefan Koch, editor. *Free/Open Source Software Development*. Idea Group, Inc., 2004.

[14] J. Kovse, T. Härder, N. Ritter, Steiert H.-P., and W. Mahnke. Supporting collaborative authoring of web content by a customizable resource repository. In *Proceedings GI/OCG-Jahrestagung Informatik 2001, Workshop "Web Databases" (Wien)*, pages 358–367, September 2001.

[15] Christoph Lameter. Debian gnu/linux: The past, the present and the future. In *Free Software Symposium 2002*, October 2002.

[16] M.A. Mark and J. Greer. Evaluation methodologies for intelligent tutoring system. *Journal of Artificial Intelligence in Education*, 2/3(1):129–154, 1993.

[17] R. Mazza and V. Dimitrova. Visualising student tracking data to support instructors in web-based distance education. In *Proceedings of the 13th International World Wide Web Conference (WWW 2004) - Alternate Educational Track*, pages 154–161, May 2004.

[18] S. Noel and J-M Robert. Empirical study on collaborative writing: What do co-authors do, use, and like? *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 13(1):63–89, 2004.

[19] MIT OpenCourseWare. `http://ocw.mit.edu/index.html`.

[20] Harvard University Library Open Collections Program. `http://ocp.hul.harvard.edu/`.

[21] Eric S. Raymond. The cathedral and the bazar. *First Monday*, 1997. `http://www.firstmonday.dk/issues/issue3\_3/raymond/`.

[22] Christian Robottom Reis and Renata Pontin de Mattos Fortes. An overview of the software engineering process and tools in the Mozilla project. In *Workshop on Open Source Software Development*, February 2002.

[23] Bordin Sapsomboon, Restiani Andriati, Linda Roberts, and Michael B. Spring. Software to aid collaboration: Focus on collaborative authoring. Technical report, School of Information Sciences, University of Pittsburgh, 1997. `http://www.sis.pitt.edu/~spring/cas/cas.html`.

[24] George Siemens. Open source content in education: Developing, sharing, expanding resources. *Pitch: peer reviewed online journal in Instructional and Learning Technology*, 2003.

[25] Utah University Open Learning Support. `http://ols.usu.edu/`.

[26] Luca Triacca, Davide Bolchini, Luca Botturi, and Alessandro Inversini. Mile: Systematic usability evaluation for e-learning web application. In *Proceedings of ED-MEDIA04 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Lugano, Switzerland, June 2004.

[27] Guido van Rossum and Jr. Fred L. Drake. *The Python Language Reference Manual*. Network Theory Ltd, September 2003.

[28] Riina Vuorikari. Why europe needs free and open source software and content in schools. *INSIGHT*, April 2004.

[29] Wikipedia. `http://en.wikipedia.org/wiki/Main_Page`.