

Edukalibre: a tool for collaborative creation of educational material

Jesús G. Barahona, Luis Lopez, Teófilo Romera, Luis Cañas

{jgb,llopez,teo,lcanas}@gsyc.escet.urjc.es

Grupo de Sistemas y Comunicaciones (GsyC). Universidad Rey Juan Carlos.
Escuela Superior de Ciencias Experimentales y Tecnología.
c/ Tulipán s/n. 28933 – Móstoles (Madrid)

Libre Software with its characteristics and its particular development model, has promoted the birth of a set of new working methodologies which has shown as very useful in several fields. In this approach, the Edukalibre project tries to import and adapt those methodologies to the problems binded to the creation and management of teaching materials. In this project, we have developed a system which provides users with mechanisms for the collaborative authoring of documents with the following advantages: 1) It allows several authors to collaborate in the authoring of a document, 2) Authors can keep track of a document's history and have access to different versions through the document's life, 3) Users are free to choose from a wide range of editors to create contents for the system documents, or even edit them on-line within the system 4) There is also freedom to select the tool for the interaction with the system (alternatives range from web browsers to WebDAV clients), 5) Automatic conversion to and from a wide variety of formats (editable formats like docbook, OpenOffice and latex, printable formats like PDF and postscript or HTML ready to be published in the web). There is a fully functional version of this system and its development has been entirely based in Libre Software tools and is as well distributed under a Libre Software license.

Keywords e-learning; collaborative edition; version control systems; libre software; open source

1. Introduction

Information and Communication Technologies (ICT) are having an important impact in all aspects of human life. Education is one of the fields where this impact is deeper. For this reason many institutions, organizations and companies are exploring the possibility of incorporating e-learning technologies into their educational processes [1,2,3]

Currently, most educational institutions use this kind of systems for re-enforcing or complementing other traditional teaching techniques. Hence, e-learning applications adapt to traditional models which are mainly based on a clear distinction between instructor and students, each of whom have a pre-established and well defined role. This is why the vast majority of educational web-based systems are conceptually designed for the elearning-by-ereading model and are build as large monolithic structures fixed in length, sequenced and scoped, non-flexible for continuous updates, difficult to organize, non-reusable or hard to reuse, etc [4].

Novel methodologies and architectures are being investigated with the purpose of providing more effective mechanisms for the creation, sharing and reuse of educational resources on the web. In particular, there is a need for innovative self-controlled and self-paced learning techniques based on the learning-by-doing model of education in a collaborative environment. In this direction, our proposal consists on applying a set of methodologies resembling those used in the Libre (free, open source) Software projects to catalyse this changes.

The Libre Software development methods have been used for decades changing the way software is produced and deployed [5,6]. Their methodologies are mainly based on the existence of strong communities of practitioners who share experiences, code and knowledge and constantly help each other.

We describe a novel, truly open platform to support the creation of free, collaborative constructed educational content on the web, which has been deployed 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 program¹. The project started on October 2003 and is expected to last until December 2005. It is coordinated by University Rey Juan Carlos (Spain), and includes partners from University of Leeds (United Kingdom), University of Porto (Portugal), University of Karlsruhe (Germany). The project web site is <http://www.edukalibre.org>.

We wish to remark that the main focus of this project is on the problem of collaborative educational content creation. For this reason, the Edukalibre system cannot be seen as a full featured Learning Management System (LMS), covering the whole learning process, as the ones described by popular standards like SCORM.

The Edukalibre System is described in the rest of this paper. Firstly we will introduce the Edukalibre System architecture. Then a brief description of all its capabilities will be enumerated. The rest of the document will show how the system works with some real life use cases.

2. Edukalibre System Architecture

The Edukalibre System is composed of a set of modules and each of them is specially devoted to a specific functionality. These modules are kept as independent from each other as possible. The system architecture (Fig. 1) is divided into two layers. The lower one (Core Level) which holds up the rest of the application and provides the basic functionality storing the documents and makes automatic format conversions with them. The higher layer (Client Level) which is composed of various interfaces to the system which enables users to interact with it.

2.1 Edukalibre Documents

Documents are the functional unit in which the whole system is based. Documents classify in two categories once they get to the system. They can be *base documents* or *final documents*.

- Base Documents: These are documents that can be modified by users. Currently supported formats for *base documents* are: Docbook, LaTeX and OpenOffice.
- Final Documents: These other are obtained from the *base documents* automatically, being this one of the main functionalities of the Edukalibre System. Some of the final formats a document may appear in are: Postscript and PDF adequate for printing; HTML for web publishing; plain text which is highly portable.

2.2 Lower Layer: Core Level

This layer is composed of various modules that work together but remain independent from each other. These modules provide the main functionalities of the system. Here is what each of them does:

- Base documents repository: This is the main module in this layer and it is based in the Subversion version control system. It hosts the base documents and keeps track of their history, permissions and correctness.
- Final document repository: Based on an Apache HTTP server, this second module provides a repository for the *final documents* of the system so that the upper layer can access them and make them available to users.
- Format conversions: This is a key module in this layer since it provides one of the main functionalities of the Edukalibre System. This module generates automatically *final documents* from *base documents* each time the latter are created or modified.

¹http://europa.eu.int/comm/education/programmes/socrates/minerva/ind1a_en.html

- Information extraction: This module obtains information from the documents held in the system, such as title, authoring, creation and modification dates, etc. and makes it available to the upper layer.

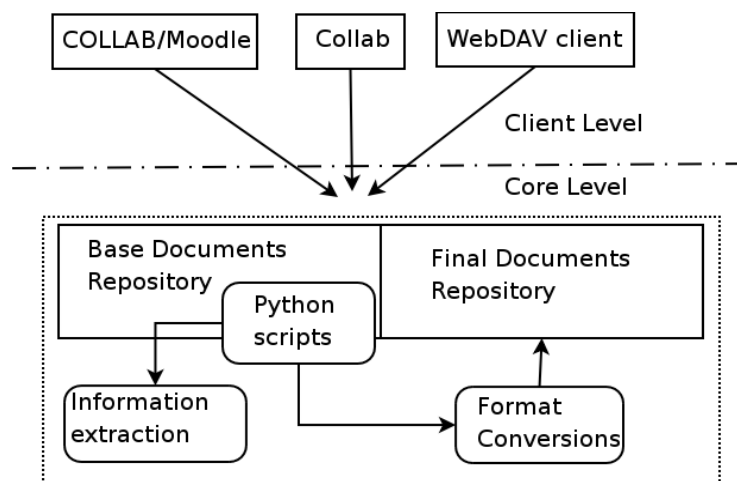


Fig. 1 Edukalibre System architecture overview

2.3 Upper Layer: Client Level

The Edukalibre System can be accessed by users from different interfaces. These interfaces reside in an upper architectural level that we call *client level*. Some of them have been developed by the Edukalibre Team and others are generic standard interfaces. Some of these interfaces are:

- COLLAB: This is a PHP-based interface which can be integrated in a simple manner in any web-portal. It supports authentication, creating and uploading new documents and accessing documents already stored in the system to update them, retrieve information about their history or even edit them on-line. All of this, just by using a web browser.
- HTTP Access: Read only access to both base and final document repositories is provided through HTTP protocol. Any web browser would do for this.
- WebDAV Access: WebDAV extends HTTP to provide some useful functionalities. With WebDAV users can access files in the system as if they were in their local file system.

3. Edukalibre System features

This section will briefly introduce the main Edukalibre System features and benefits.

- Collaborative edition: One of the main features of the Edukalibre System is the possibility of collaboratively create, elaborate and manage documents. This means that the system supports several different authors working in the same document.
- Version control: When various authors work in the same document there is a strong need for a strict version control. Edukalibre provides authors with mechanisms to review the history of a document, compare different versions of the same document to look for changes, undo changes, creating new branches of a document and even merge those branches in the future.
- Different edition styles: The Edukalibre System tries to manage documents created with as many different editors as possible. Anyone could use WYSIWYG editors (like OpenOffice), text editors (from VI to Emacs) or even on-line edition from COLLAB (with a simple web browser) to modify documents in the system.

- Automatic format conversion: The system will obtain final documents from base documents without much effort from users. This process is automatic and fully transparent.

4. Use Cases

4.1 Uploading a document through COLLAB

There are two options to create a new Edukalibre document with COLLAB. The first one is letting the system place an empty Docbook template in the repository for you. The second one is to upload your own document through the web form in COLLAB.

1. Interface: To upload a document through COLLAB you should use the form located in the right box of the main page .

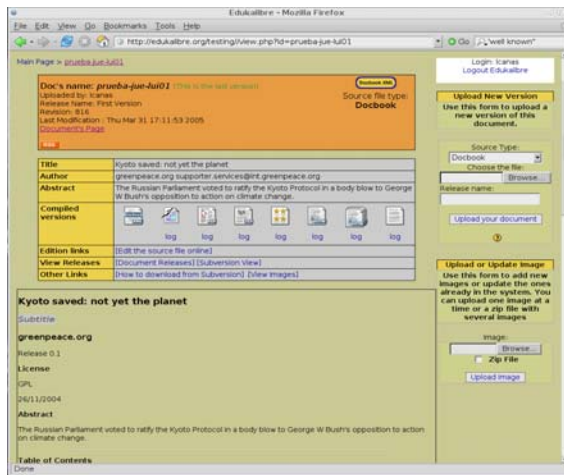


Fig. 2 COLLAB: document page.

document page (Fig. 2). This page contains summarized document information including author, title, abstract, and modification dates. It also contains links to final documents generated from the initial base document.

4.2 Updating a document with COLLAB

You can use COLLAB to update a document in two ways. It depends on if you use COLLAB's built-in on-line editor to modify the document or if you prefer to download the base document from the system, use your own editor to modify it, and then upload it again to the system.

1. Interface: You should start in the document page (Fig. 2).
2. Actions:
 1. On-line edition
 1. Click on *edit the source on-line*. This will take you to the on-line edition page
 2. Once you have done you changes, click on *update document*
 2. Download the base document
 1. Get the source from the document page
 2. Modify it in your system as you would with any other file

2. Actions:

1. Create a new document from a Docbook template
 1. Click on the checkbox located in the right form
 2. Type a name for your document (this name will be used as an ID for your document)
 3. Click on *upload your document*
 2. Create a new document from your own file
 1. Type a name for your document (this name will be used as an ID for your document)
 2. Select the document type (OpenOffice-native, OpenOffice-Docbook, Docbook or latex)
 3. Select your file
 4. Click on "upload your document"
3. Results: Once your document has been uploaded, you will be redirected to the

3. Upload it again to the Edukalibre System
3. Results: Just like when you upload a new document, the system will generate a new set of final documents from the modified base document. Once again you will be then redirected to the document page.

4.3 Review the history of a document with COLLAB

1. Interface: You should start in the document page at COLLAB (Fig. 2).
2. Actions: COLLAB can show the history of a document in two ways.
 1. To view previous document releases
 1. Click on “view releases”
 2. To view a repository based view
 1. Click on “subversion view”
3. Results: Those links will present two different pages. The first one will show a new page with summarized information for every previous release of the document. It is possible to get to a document page (like the one shown in Fig. 2) for each of them. The second option uses ViewCVS to present a Subversion view of the history of the document. This is more useful to keep track of changes for every single file of the document.

References

- [1] H.H. Adelsberger, B. Collins, and J.M. Pawloswski. *Handbook on Information Technologies for Education and Training*. Springer, 2002.
- [2] J.M. Rosenberg. *E-Learning*. McGraw-Hill, 2001.
- [3] W. Horton. *Designing Web-Based Training*. Wiley Computer Publishing, 2000.
- [4] V. Uskov. A 3rd generation web-based instructional tool for education and lifelong training. *International Journal of Continuing Engineering Education and lifelong Learning*, 13(1/2):110-131, 2003.
- [5] Nikolai Bezroukov. Open source software development as a special type of academic research. *First Monday*, 4(10), October 1999.
- [6] 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*, Edinburgh, Scotland, UK, May 2004.