

Transferring Libre Software Development Practices to the Production of Educational Resources: the Edukalibre Project

Jesús M. González-Barahona
Universidad Rey Juan Carlos
Mostoles, Spain
jgb@gsync.escet.urjc.es

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

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

Diego Chaparro
Universidad Rey Juan Carlos
Mostoles, Spain
dchaparro@gsync.escet.urjc.es

Teo Romera
Universidad Rey Juan Carlos
Mostoles, Spain
teo@gsync.escet.urjc.es

Abstract – The transfer of methodologies common in libre (free, open source) software development to the domain of educational resources can radically change the way educational content is developed and used, enabling both educational practitioners and students to become actively involved in its creation and distribution. New software architectures and tools are needed to effectively support this process. This paper describes a platform aimed to support the creation of free, collaboratively constructed educational content on the web, which has been developed within the Edukalibre project. It provides easy access to core technologies: 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.

I. INTRODUCTION¹

Most educational institutions, and especially universities, have been exploring for several years 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 could be deployed in a variety of educational paradigms.

In this line, this paper discusses how to transfer some of the practices usual in libre (free, open source) software development [12] to the educational domain. Strong communities of libre software practitioners who share experiences, methods, code, and constantly help each other have been built in a number of domains, ranging from operating systems to desktop environments, web browsers, web servers, office suites, and many other [3].

Following this idea, we propose a novel platform to support the creation of free, collaboratively constructed educational content, which has been developed within the Edukalibre project. 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 (Germany), and University of Lugano (Switzerland)².

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. Finally, in the conclusions, we will sketch some plans for future work.

II. 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 collaboratively building programs, which is sometimes 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, some issues are found in most libre software projects:

- Frequent and early release, even when the software is not fully tested. Eric S. Raymond spells this out as “Release early, release often” [10].
- Quality by inspection of many individuals, including those outside the group of developers.
- Development done asynchronously by geographically distributed groups, and only using software tools to coordinate[3].

¹The 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.

²Edukalibre website: <http://edukalibre.org>

- Mixture of voluntary and paid work

The Edukalibre project is exploring how this model can be applied to education, and specifically to the creation and maintenance of educational materials. In particular, the project is focused in an environment with the following assumptions:

- Educational materials will be located mainly on the Internet.
- Materials will be produced by groups of educators, usually in different institutions, and geographically dispersed.
- Materials will also be used, commented, and maybe modified by students.
- Educators and students will need simple to use, yet powerful, tools to be able to collaborate in the way libre software developers do.
- The public availability of produced materials will enable the collaboration of third parties.
- Several licensing terms are possible, including free distribution, but also for profit publication by publishers.

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.

III. 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, 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*. The document repository is built on top of a *version controlled storage system*, with document conversion and management tools written in Python. 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.

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 version control system or the Moodle Learning Management System.

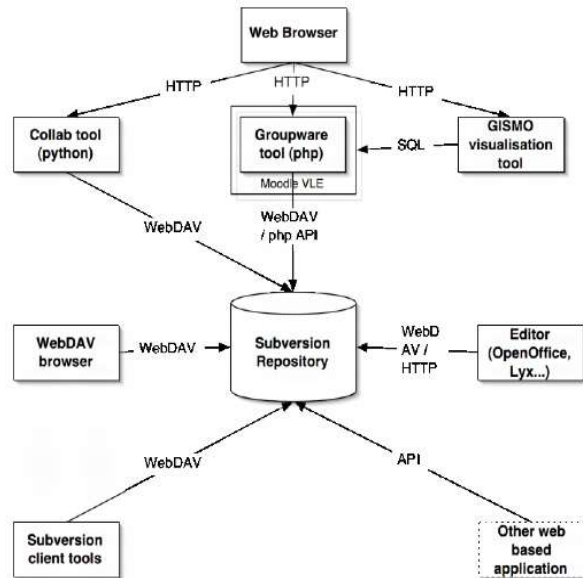


Figure 1 Architecture of the system

But those components 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.

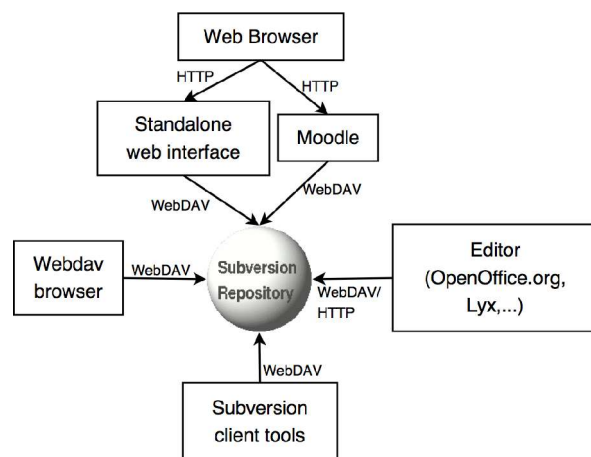


Figure 2 Components of the collaborative editing tool

A. 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. Figure 2 shows the main components of the collaborative editing subsystem and their relationships. These components are:

- *Version control system*: Every document created has a main version and a history of older revisions. Currently, the platform uses Subversion for this task.
- *Editors and word processors*: The project does not intend 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. For this matter, support for OpenOffice.org is provided in two ways: as a native format, and with conversion tools to/from DocBook/XML. However, any text editor can be used to modify the documents of the system, if it can manage either DocBook/XML or LaTeX (also a supported format).
- *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 the conversions from XML formats are performed using XSLT stylesheets. The XSLT processors used are xsltproc and xalan. Conversions to PDF and Postscript are done using TeX/LaTeX and Ghostscript.
- *Web interfaces*: One of the goals of the project is to allow for as many 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 are already provided like a collaborative editing system (COLLAB), which is a simple Python-written web interface. There is also a PHP groupware application written as a component of Moodle, called ConDOR which aim is to examine the effectiveness of the tools within a simple groupware application.
- *Communication protocols*: Standard protocols are used to connect the different components. They are the ubiquitous HTTP [1], and WebDAV [4] (an extension to HTTP tailored to the needs of collaborative editing and management of files on remote web servers).

The Edukalibre system offers a PHP API to allow the easy integration of any PHP application with it. The Edukalibre PHP API builds on PEAR::SVN to provide simple XHTML form generation for interacting with the repository and tools for folder

creation, file deletion, history extraction and functions that return the differences between certain versions of a given document.

B. Formats

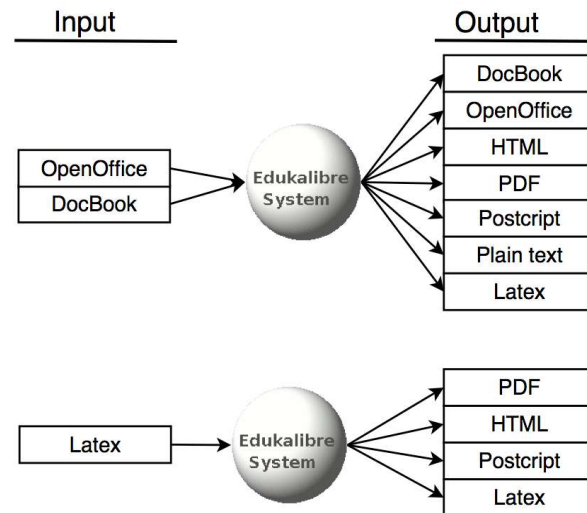


Figure 3 Document formats

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 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.
- End-user formats, which are automatically generated by the system from the base formats. These include PDF or Postscript, XHTML in several flavors, or plain text.

C. Repository Manager

The repository manager is the core of the Edukalibre platform, being in charge of storing the documents, creating the end-user formats, and providing information about the documents stored in the system. The core repository is a storage system containing documents, together with information about them.

The document conversion and metadata extraction system provides additional functionality to the repository manager. The 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 validates the file, stores the file in the subversion repository, and converts the file to the end-user formats. At the end of the process, some information is extracted from the document, and 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).

IV. 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 that 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 as a whole allows for the easy modification of the documents by educators, or even by students (if they have the required permissions), in a wiki-like fashion. 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.

The Edukalibre platform has been extensively tested and case studies are available on request. The Edukalibre platform is in use across six academic institutions, and comprehensive testing and evaluation results are available, outside of the context of this short paper.

Currently, the Edukalibre project is in its second (and last) year, when in addition to some further evaluation activities, new lines of work are being explored. 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 a better 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 or Zope), 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 provide support and hosting for libre software projects.

V. REFERENCES

- [1] R. Fielding et al, 1999, Hypertext transfer protocol - HTTP/1.1, RFC 2616. *Technical report, Network Working Group, IETF.*
- [2] BSCW: Basic Support for Cooperative Work, <http://bscw.fit.fraunhofer.de/>.
- [3] Daniel M. German, 2003, The GNOME project: a case study of open source, global software development, *Software Process Improvement and Practice*, pages 201-215
- [4] Y. Goland et al, 1999, HTTP extensions for distributed authoring - WEBDAV, RFC 2518. *Technical report, Network Working Group, IETF.*
- [5] Connexions: Sharing knowledge and building communities. <http://cnx.rice.edu/>.
- [6] J. Kovse et al, 2001, Supporting collaborative authoring of web content by a customizable resource repository. *Proceedings GI/OCG-Jahrestagung Informatik 2001, Workshop "Web Databases" (Wien)*, pages 358-367
- [7] M.A. Mark and J. Greer, 1993. Evaluation methodologies for intelligent tutoring system. *Journal of Artificial Intelligence in Education*, Vol. 2/3, No. 1, pages 129-154
- [8] R. Mazza and V. Dimitrova, 2004, Visualising student tracking data to support instructors in web-based distance education. *Proceedings of the 13th International World Wide Web Conference (WWW 2004) - Alternate Educational Track*, pages 154-161.
- [9] R. Mazza, C. Milani. GISMO: a Graphical Interactive Student Monitoring Tool for Course Management Systems. *T.E.L.'04 Technology Enhanced Learning '04 International Conference*. Milan, 18-19 November 2004.
- [10] Eric S. Raymond, 1997. The cathedral and the bazaar. *First Monday*, http://www.firstmonday.dk/issues/issue3_3/ramond/
- [11] Luca Triacca et al, 2004, Mile: Systematic usability evaluation for e-learning web application. *Proceedings of ED-MEDIA04 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Lugano, Switzerland.
- [12] Ahmed E. Hassan et al, 2001, Software engineering research in the bazaar. *Proceedings of the 2nd Workshop on Open Source Software Engineering at the 24th International Conference on Software Engineering*.