

Fred de Vries,
Graham Attwell,
Raymond Elferink,
Alexandra Tödt (Eds.)

Open Source for Education in Europe Research & Practise

Conference proceedings

Heerlen, The Netherlands, November 14 and 15, 2005



OpenUniversiteitNederland

Open Source for Education in Europe, Research & Practise

Conference proceedings

Heerlen, The Netherlands, November 14 and 15, 2005

Fred de Vries,
Graham Attwell,
Raymond Elferink,
Alexandra Tödt (Eds.)

First Published 2005 by
Open University of the Netherlands, Heerlen
Educational Technology Expertise Centre

Cover design
Jeroen Storm

Printed and bound by
Grafisch Centrum, Open University of the Netherlands, Heerlen

CIP information
Koninklijke Bibliotheek, Den Haag

Open Source for Education in Europe, Research & Practise
Conference proceedings
Heerlen, The Netherlands, November 14 and 15, 2005
by Fred de Vries, Graham Attwell, Raymond Elferink, Alexandra Tödt (Eds.)



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5
Netherlands License: <http://creativecommons.org/licenses/by-nc-nd/2.5/nl/>

ISBN 90 358 2318 4
NUR 840

Table of Contents

Introduction	7
--------------------	---

Theme 1 - Open Standards, Architectures and Applications

Open Standards, Sustainability and Open Source Software	
<i>Alexandra Tödt</i>	11
Tools for Enabling Conformant e-Learning Specification Implementations	
<i>Owen O'Neill, Rob Nadolski</i>	17
Open Source Flash Projects The Way Forward For Reusable Learning Objects	
<i>Catherine Hurley, Timothy Hall</i>	21
e-Portfolios / PDPs: LUSID Developments And Tool Deployment In Bodington	
<i>A.C. Marshall, S.Grant, J. Strivens, S. Mehan</i>	25
Practical pedagogical uses of IMS Learning Design's Level B	
<i>Daniel Burgos, Rob Koper</i>	33
Metadata and Application Profiles: a Data Model	
<i>Robert Schuwer, Henk Hangyi, Darco Jansen</i>	41

Theme 2 - Sustainability Strategies - Managing Open Source

How a FIRM (Flexibility, Innovation, Robustness, and Maturity) Argument for FOSS (Free and Open Source Software) Can Displace FUD (Fear, Uncertainty, and Doubt)	
<i>Derek Briton</i>	47
Overcoming Proprietary Hurdles: CPOINT as Invasive Editor	
<i>Andrea Kohlhase</i>	51
From learning design model to automated synthesis of a web-based educational application: A CASE tool	
<i>Symeon Retalis, Andreas Papasalouros, Vaggelis Skouteris, Kostantinos Siassiakos</i>	57
Open Source Software in Teaching Physics: A Case Study on Vector Algebra and Visual Representations	
<i>Erdat Cataloglu, Mustafa Baser</i>	65
Using IMAP to Build a Virtual Learning Environment	
<i>Michael Klebl</i>	75
KEILab (Knowledge Exchange Initiative Laboratory): a research project about open source and elearning	
<i>Mario Vercellotti, Eleonora Pantò, Matteo De Simone</i>	87
Comparing Software Development Models: Structural Problems in the Cathedral and Bazaar metaphors	
<i>Raymond Elferink, David Griffiths, Sara Zondergeld</i>	91

Theme 3 - Open Content - Issues and Implications

Openness as an evolutionary determinant of human existence	
<i>Wim Westera</i>	97
Collaborative Development Strategies for Open Source Involving the Users' Perspective	
<i>Stefanie Panke, Christian Kohls</i>	107
E-LANE: Open Source eLearning in Latin America	
<i>Dudley Dolan, Carlos Delgado Kloos, Mario Muñoz Organero, Abelardo Pardo</i>	117
Edukalibre Project: Versatility in e-learning	
<i>Jaime Villate, Guilherme Dutra, Nuno Faria, Mário Lopes</i>	127
OMDoc: Open Mathematical Documents	
<i>Michael Kohlhase</i>	137
Open Content and Source: European Schoolnet Riding the Wave	
<i>Riina Vuorikari, Karl Sarnow</i>	145
Helping to develop an Open Source Curriculum: the case of TOSSAD E.U. funded project	
<i>Michele Marchesi</i>	157

Theme 4 - Learning with Open Source

Building Eclectic Personal Learning Landscapes with Open Source Tools	
<i>Marco Kalz</i>	163
Adapting Open Source Software for Education: Challenges, methodologies and results	
<i>Jim Henderson, Evan Brown, Chris Mitchel</i>	169
Hamlet goes XML: CrossAnnotationLinking and Personal Learning Experiences	
<i>Benjamin Birkenhake, Stefanie Panke, Andres Witt</i>	179
Implementing and Adjusting Open Source Solutions: the Internet Seminar "Introduction to Applied Computational Linguistics"	
<i>Karin Naumann, Lothar Lemnitzer</i>	189
Towards the open cross-disciplinary research environment	
<i>Ernesto Damiani, Paul G. Mezey, Paolo M. Pumilia, Anna Maria Tammaro</i>	197
Stud.IP – High Performance Support for Lecturing and Learning	
<i>Marco Bohnsack</i>	207

Conference Open Source for education in Europe, Research and practice

Introduction

Open Source Software (OSS) is of increasing interest in education. It can provide better quality software, increased pedagogic choice, enhanced flexibility and new business and social models. Furthermore, the idea of Open Source is being extended into other areas including the production of e-learning materials.

This conference on Open Source for education in Europe – Research and Practice has been organised by two European eLearning Initiative sponsored projects, SIGOSSEE and JOIN, which have been investigating the potential of Open Source and providing services for educational institutions wishing to implement Open Source products, together with the Open University of the Netherlands (OUNL) which has a long track record of innovation in e-learning.

Last but not least, the event is sponsored by the Swiss International Relations and Security Network.



Despite – or because of – the growing interest in Open Source Software there is an increased focus on a number of issues related both to Open Source in general and to the adoption and application of Open Source in education.

These issues are of particular interest to the work of the SIGOSSEE project which has undertaken studies on:

- a) Standards and architectures for OSS
- b) User requirements and usability issues in the development of OSS
- c) Social, cultural and legal issues in OSS
- d) Organisation and management issues, sustainability and support infrastructural needs for OSS

The SIGOSSEE project has also been on take-up of OSS in selected institutions over the lifetime of the project, in order to determine best practice in different environments and the JOIN project has developed a directory of Open Source Virtual Learning Environments.

The studies undertaken through the SIGOSSEE and JOIN projects have informed the selection of themes for this conference and the grouping of papers included in this conference proceedings.

The papers address major issues in Open Source in education in Europe including:

- Learning with Open Source
- Open standards for e-learning
- Sustainability strategies – managing open source
- Open content, issues and implications

Conference Themes

Learning with open source

e-learning requires the development of new pedagogic approaches to teaching and learning. Open source has the potential to provide for pedagogic diversity and choice. Everyday applications such as web logs and wikis are increasingly being used for learning. Developers need to work together with teachers, trainer and learners in developing new applications. This strand will examine different pedagogic approaches and consider their implications for the future of e-Learning.

Papers in this strand address the following issues:

- Personal learning environments
- Social software and learning
- Collaborative development strategies
- Informal learning and its implications for future e-Learning applications

Open Standards for e-learning

Open standards are critical for developing interoperability of systems and architectures for e-learning. Open standards can allow the development of component architectures and allow the exchange of data between different applications. This facilitates the implementation of different pedagogic approaches. It also permits easy upgrading of systems. However even with open standards there remain problems in how the quality of e-learning software is measured and assured.

Papers in this strand address the following issues:

- Developing and adopting open standards and specifications
- Tools and toolkits to support open standards
- Systems and frameworks for quality assurance

Sustainability strategies – managing open source

Sustainability is a major issue in e-learning. Open Source can potentially solve this question, through standards compliance and portability of data and through robust and cheap to implement architectures.

Nevertheless there are significant issues in migrating to open source – not least of which is support and maintenance.

New business models are emerging around Open Source, However there remain issues in how open source software can be supported and maintained.


Papers in this strand address the following issues:

- Maintenance and support for open source software
- Migrating to open source software
- Developing institutional, local and regional strategies for open source
- Professional development for open source
- Models for supporting and maintaining Open source software

We were both surprised and delighted by the response to the call for papers for the conference and had considerable difficulties in selecting which papers to include in the conference agenda and proceedings.

We were concerned to address the issues and themes of the original projects but also to reflect the innovation and imagination of the education and Open Source communities. We also wished to maintain the standards of a scholarly conference and to provide a space for genuine knowledge exchange and sharing. Finally we were concerned to strike a balance between research and practice and more ambitiously to draw links between the two.

This was an ambitious endeavour and how far we have succeeded can only be evaluated by the users – the participants in the conference and the readers of these proceedings. We would like to thank all those who have contributed to the organisation of the conference, to all those who have participated in the projects and to all the conference participants. Finally we acknowledge the support of the European Commission eLearning Initiative.

 e Learning

 Education and Culture



Fred de Vries,
Graham Attwell,
Raymond Elferink,
Alexandra Tödt

Open Standards, Sustainability and Open Source Software

Alexandra Tödt
Research Institute for Political Science and European Affairs
University of Cologne
alexandra.toedt@uni-koeln.de

Abstract

This paper outlines the concepts of 'open standards' and 'sustainability'. Both concepts are key issues in the field of e-learning. Do we need standard-based e-learning? What kinds of standards do we need and why? An overview over existing 'open standards' in e-learning will be provided. As well I will outline how open standards and the sustainability of an investment in technology and organisation of e-learning efforts are interconnected. What secures this investment? Because open standards decrease dependency on certain product. How can this be achieved? Users get independent of a certain product if the system in use is standard-conformant and there are other systems to be switched to. Particularly suitable for e-learning is content that does not change regularly but is settled and thus can be re-used. The time it takes to prepare the content and the pedagogical scenario is well invested if one can re-employ it properly.

1 What are open standards?

There is a lot of arguing about what cases are properly labelled as 'open' or 'standards'. We distinguish between 'de facto standards', 'de jure standards' and 'open standards'.

'De facto standards' are proprietary specifications which implementations everybody uses. De facto standards have not been approved by a standard setting body. A well-known example of a de facto standard in the content producing realm of e-learning is the MS Word Document. Every end-user has an advantage of employing a de facto standard, since it enhances his or her ability to share data.

A 'de jure standard' is an approved specification. It explicates a degree of excellence required for a particular purpose. This standard is formally accepted to serve as a basis for implementation. Standards can be characterised as a quality yardstick approved by a specialised institution, a standards body. De jure standards are to be preferred because they are public and have documentation. Proper standards are ratified by an international standard setting body like IEEE or W3C.

An 'open standard' is a specification for achieving a specific task. What earns a standard the attribute 'open' is the practice of offering and operating the standard. The goal of open standards is to create a fair, competitive market for implementations of the standard.¹

¹ The World Wide Web Consortium (W3C) is an example of a standards-setting body that has recently taken a position on patents to align their definition of a Web standard with this definition of an open standard.

To qualify for being commonly accepted as an 'open standard', this must satisfy the following criteria²:

- It must be available for everybody in order to read and implement it. This requires the standard's text and reference implementations to be available for download, preferably at a low cost or no costs at all. The licence attached to the standard's documentation must allow free choice of license for the implementation.
- It allows various implementations of the standard and does not tie end-users to certain vendors.
- An open standard must allow a wide range of implementations.
- The standard can be implemented without paying a royalty or a fee. If the standard contains patented code, it must be licensed in a royalty-free and non-discriminatory way. Still, the certification of compliance by the standards organization may involve the payment of a fee. Open Standards, however, allow to spread best practices since they are royalty free.
- No implementation must be discriminated for any reason other than the technical standards compliance.
- Implementations of Open Standards may be extended, or offered in subset form. The standards' organisation is not required to certify subset implementations, and is allowed to place requirements upon extensions.
- It avoids predatory practices. Open Standards may employ license terms that protect against subversion of the standard by embrace-and-extend tactics. The licenses attached to the standard may require the publication of reference information for extensions, and a license for all others to create, distribute, and sell software that is compatible with the extensions. An Open Standard may not otherwise prohibit extensions.³

There are some arguments about the appropriate specifications the term Open Standard should be restricted to. Standards published by ISO for example are open but may require patent licensing fees for implementation.

1.1 List of open standards relevant to e-learning

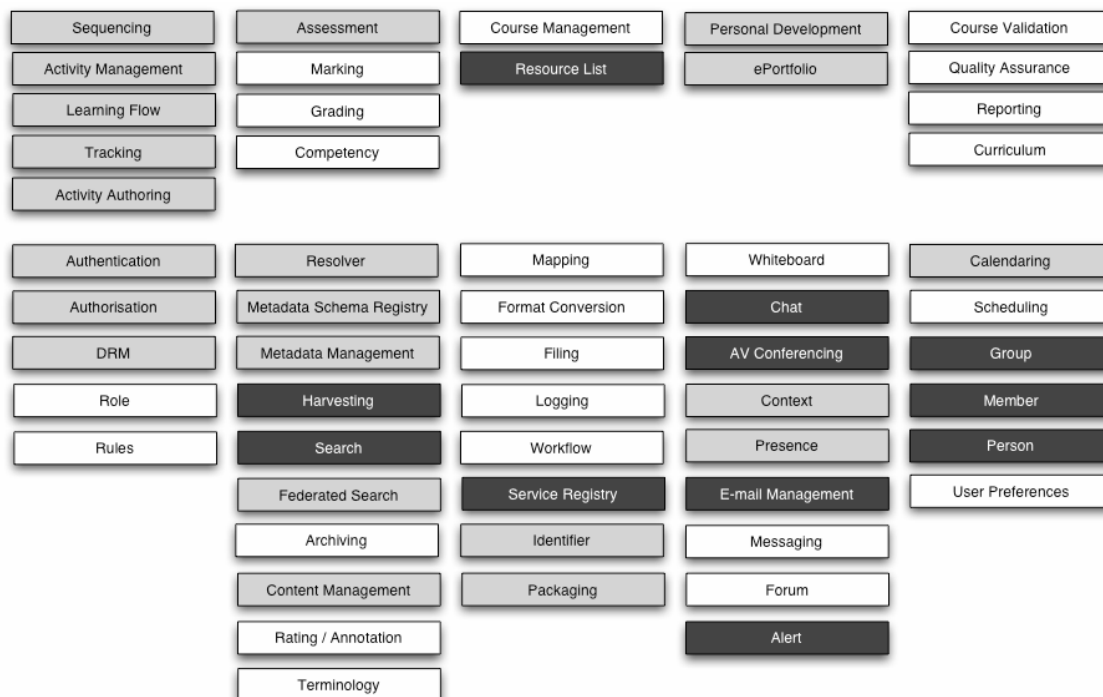
To produce a complete list of existing open standards would consume more space than available. So I just highlight some of those that have been really successful and are applied ubiquitously. These comprise: DLL, HTTP, SOAP, SIP, SNMP, VoiceXML, SALT, SAPI, VoIP, ethernet, HTML/XHTML, SQL, IP, TCP, PDF, OpenDocument.

The diagram following displays the services needed for e-learning according to the JISC. All services that have some form of open standard are highlighted in light grey. Where a complete set of open standards – including both data models and service definitions – is available, the service is highlighted in dark grey with white text.⁴ Some changes might have occurred if the OKI OSIDs have been submitted to an open standards process.

² See: Sender ID: A Tale of Open Standards and Corporate Greed? At <http://www.circleid.com>

³ See: Open standards - Principles and practice. At <http://perens.com/OpenStandards/Definition.html>

⁴ See: JICS e-learning framework. At <http://www.cetis.ac.uk/members/enterprise/frameworks>



Open standards are an important sustainability issue

Systems theory – a certain type of social science theory – posits forward the concept of sustainability. It describes an approach to balance today’s needs with tomorrow’s choices. Imminent demands must not be allowed to ruin the indefinite future. This section will discuss the pros and cons of open standards with regard to sustainability.

2.1 Pros of standards

Standards avoid the most dangerous characteristics of the information-based life format: The lock-in effect. “First and foremost, standards enhance compatibility, or interoperability, generating greater value for users by making the network larger.”⁵ Standards expand available network externalities. Interoperability creates substantial consumer benefit, since enhanced ability to share data attracts still more users. Interoperability allows for synergies coming from users creativity. Thus choosing software that is open standard compliant reduces technology risks.

This is due to the factors listed below⁶:

- Network effects: The benefits of open standards result in a network effect, since open standards are supported by several suppliers normally.
- Durability: The very nature of being standards-compliant adds to the durability of every implementation.
- Choice / Independence: The user is not subject to a single company’s strategy.
- Variety: the user can choose between different implementations of the standard and different vendors. He or she can choose the implementation and / or vendor most appropriate for his or her needs. In addition, users can make realistic threats to change the supplier. Competition increases the product quality. This allows covering a vast range of deployment scenarios and quality needs.
- Interoperability: “Interoperability is achieved when components are able to function together to share in the fulfilment of a process. [...] Increasing interoperability increases your ability to connect and automate processes that transcend technologies, platforms, languages and customizations. Open standards and open architectures are

⁵ Shapiro, Carl: Information rules, 1999, p. 229

⁶ See: Business Case for Open Standards By Erik Sliman. At <http://www.openstandards.net>

continually improving to reduce the barriers to integration of disparate systems.”⁷ If systems are compliant to open standards you and your partners in teaching and learning can exchange data.

- Lower Total Cost of Ownership: Switching costs are much lower. Changing vendors or implementation is less costly. Documented interfaces lower integration workload that may be needed to make software work together. Investment in software will yield higher returns over a period of time if one invests in open standard software.

Defined and documented interfaces ease integration as well as interoperability. Implementation of standards increases compatibility. Therefore, compliance to open standards is an advisable strategy to protect one’s investment over a long term of time. “If the standard is truly open, consumers will be less concerned about lock-in. They can count on future competition.”⁸

2.2 Cons of standards

“In principle, standards would be beneficial both to designers as well as to end users of any product or service.”⁹ But there are shortcomings of standards are to be mentioned:

- The standard-setting procedure is too slow and too political.
- It takes ages to formulate some specification that can be agreed upon. For a dynamic market and the technological innovation ratification processes are not to be waited for.
- A compromise that arrived after ages of disputes may be not particularly useful because it may be outdated by reality.
- During negotiations to formulate a compromise stakeholders tend to promote solutions that serve their specific interest this can lead to useless compromise.
- In some cases the certification of compliance may be an expensive and harrasing procedure.
- Some standards are very specific and detailed beyond reason, and getting conformant to it requires a lot of work going to the system that could be used productively instead.
- Finally, by definition a standard does not evolve with user-needs. It is settled. This may be a problem.

2.3 Who wins and who loses from standards?

Standards change the nature of the information game. What do stakeholders think of standards?

- Users: “Consumers generally welcome standards: they are spared having to pick the winner and face the risk of being stranded. They can enjoy the greatest network externality in a single network or on networks that seamlessly interconnect. They can mix and match components to suit their tastes. And there are far less likely to become locked in into a single vendor.”¹⁰
- Complementors: E-learning complementors are professional producers of content like publishers. Using open standards allows them to produce content for a greater network.
- Designers of tools: Standards shift competition away from features and toward price, for the simple reason that many features are common across all brands. “So while more extensive standards lead to fewer compatibility problems and stronger network externalities, they can also reduce the ability of each supplier to differentiate its products, thereby intensifying price competition. For this very reason, consumers tend to seek more extensive standards than do suppliers.”¹¹

⁷ See: Business case for Open Standards By Erik Sliman. At <http://www.openstandards.net>

⁸ Shapiro, Carl: Information rules, 1999, p. 230.

⁹ See: Robin Good: Standards do we really need them? 26.12.2003

¹⁰ Shapiro, Carl: Information rules, 1999, p. 231.

¹¹ Shapiro, Carl: Information rules, 1999, p. 231.

Outlook and Roadmap

Network markets are of the winner-takes-all-nature; this makes cooperation more important than ever.

Public bodies push for open standards and everybody should. Employing open standards decreases a users costs and increases his or her range of choice, adopting open standards lowers risk and future-proofs software choices. "Obtaining these advantages begins with the decision to declare open standards among your highest priorities. Decide to evaluate open standard options in all your considerations. Address it in your planning. Begin to build the process of understanding, contrasting and developing conclusions of how open standards can improve your decisions and impact your business."¹²

European governments have already arrived at these conclusions: In 2004 the Belgian Council of Ministers and the Danish government approved new directives and recommendations for the use of open standards and software applications by the Federal Ministries.¹³

In order to facilitate data sharing, ICT systems must be based on open standards. Systems of these federal authorities that do not support open standards for archiving, exchanging and communication will be discontinued and data will be migrated to standard-compliant environments.

Without doubt it is difficult to unseat de facto standards. It takes a commitment from everybody, but it is nevertheless rational to go through these changes.

EU Commissioner Erkki Liikanen: "Open standards are important to help create interoperable and affordable solutions for everybody. They also promote competition by setting up a technical playing field that is level to all market players. This means lower costs for enterprises and, ultimately, the consumer." (World Standards Day, 14 October 2003) [1]

References

lists of groups that produce open standards: <http://www.open-std.org>

lists of open standards can be obtained at <http://www.ososs.nl>

Dutch policy and experience with open standards <http://www.ososs.nl>

Danish policy and experience with open standards <http://www.oio.dk>

Enterprise Technical Reference Model - Version 3.5 can be accessed at <http://www.mass.gov/Aitd>

For UK policy and experience see <http://www.govtalk.gov.uk>

Linux magazine : open source, open standards. San Francisco 3.2005, 8

¹² See: Business Case for Open Standards By Erik Sliman. At <http://www.openstandards.net>

¹³IDABC: *eGovernment News – 06 July 2004 – Belgium – Interoperability/Open Source Software.*

Tools for Enabling Conformant e-Learning Specification Implementations

Owen O'Neill and Rob Nadolski
Educational Technology Expertise Centre
Open University of the Netherlands
{owen.oneill, rob.nadolski}@ou.nl

Abstract

The rapid expansion of e-learning in recent years has also led to a wide range of open e-learning specifications becoming available. A commonly expressed goal among such specifications is the spread and development of interoperable e-learning systems and content. However, reaching this goal requires systems which are conformant to the specifications they are based on. Until recently, such conformance testing was a time-consuming and difficult task, made even more complicated by the potential for many e-learning specifications to be localised to better meet user-community needs. To fully realise the benefits of open e-learning specifications, there is a clear need for effective and low cost conformance testing of e-learning systems and content. Telcert is an EU-funded project which is developing a number of software tools and processes to enable conformance testing. This approach not only simplifies the complex task of localising e-learning specifications but also provides a method for developing test suites for conformance testing.

1. Introduction

In recent years there has been strong growth in the use and acceptance of e-learning. This growth has been paralleled by the development of a wide range of open e-learning specifications. One of the primary goals of such specifications is to enable the spread of conformant and interoperable e-learning systems and content. The process of conformance testing aims to determine whether an implementation conforms to the specification it was based on (Malek & Dibuz, 1998), while interoperability can be defined as the ability to exchange and use information across different systems (Kindrick, 1996). Conformance to specifications is a vital step towards interoperability. Many e-learning specifications permit localisations to be made during implementation, a process called Application Profiling. Application Profiles allow user communities to tailor specifications to their needs, which encourages their adoption by different communities of users. However, such localisations also further complicate e-learning specification conformance testing.

The Telcert (Technology Enhanced Learning Conformance – European Requirements and Testing) project has developed a number of tools to make conformance testing for e-learning specifications less time-consuming, expensive and complex. Telcert is a Technology Enhanced Learning research and technology development project under the European Union's 6th Framework programme, and is concerned with the development of new testing technologies to meet the diverse needs of suppliers and users of e-learning specifications. The Telcert tools enable the modification of XSD-based specifications, creation of content based on such Application Profiles, and also comprehensive conformance testing against such Application Profiles. Initial tests have been conducted with a number of IMS e-learning specifications including Content Packaging (CP), Learning Resource Metadata (MD), Learning Design (LD) and Learning Information Packaging (LIP) which have demonstrated the effectiveness of these tools when used in combination with existing conformance testing techniques (O'Neill & Nadolski, 2005).

2. Conformance Testing Approach

Use of the Telcert tools does not mandate the use of any particular conformance testing techniques. However, utilizing existing conformance testing principals facilitates the creation of efficient test suites. For example, using the Category Partition method (Ostrand 1988) the testing task can be divided in order to target potential errors more effectively. Another conformance testing technique, boundary value analysis (Ramachandran, 2003), allows the selection of values which lie along data extremes, as these are considered more likely to return an error. Boundary values include for example maximum/minimum and just inside/outside boundaries. Utilising these techniques enables the number of required test cases to be kept to a minimum, limiting the amount of test content that must be produced.

3. Tools Overview

This section provides a brief overview of the use of the Telcert tools in practice. They have been listed here in the order that they may be logically used, however this order is not mandated.

3.1 Development of an Application Profile (Schemaprof Tool)

The open source Schemaprof tool (Schemaprof, 2005) allows the creation of an Application Profile based on an XML schema file without needing to edit the schema directly. It reduces the incidence of human-error by limiting the possibility of creating incorrect or illegal modifications. Schemaprof also warns when illegal modifications are being attempted. Modifications are written to an XML file.

3.2 Creation of a Localised Schema (Schema Transformation Tool)

This tool works as a plug-in for Schemaprof, which enables the creation of a localised XML schema by incorporating the derived XML modification file and the original base specification schema. This localised schema can then form a template for the creation of content based on the application profile.

3.3 Generation of Test Content (Content Reengineering Tool)

The Content Reengineering Tool allows the creation of content based on localised XML schemas and can be used for creating test content. Using techniques such as the category partitioning method and boundary data analysis, content manually seeded with errors can be used together with error-free content to create a test suite for the Application Profile. This tool is based on the open source Reload editor (RELOAD, 2005) with features added to aid the creation and modification of content based on localised schema. Functionality has also been added to support base specification extensions.

3.4 Testing Content against an Application Profile (Telcert Test System)

In the test system, content is tested against a particular e-learning specification. For each specification, a number of tests may be derived directly from the base specification documentation. For example, the IMS CP specification contains a number of rules determining the way the package is compressed into an archive format (commonly a .zip file). If such tests are mandatory for any implementation, they may be added directly to the test system. Content can therefore be tested against the base specification (represented by the specification XSD file), the supplied application profile as produced by the Schemaprof tool, and the generic tests (derived from the specification documentation). The test system allows the creation of test sessions which link the test content with the required tests. After running a test session, the system provides detailed information on the results of each test.

4. Discussion

During initial testing within the project, the tools and processes described in this article enabled the creation of Application Profiles and simplified the process of testing against these derived Application Profiles. However, the development and testing of these tools have pointed to two particular issues which require further elaboration.

Firstly, although the Content Reengineering Tool can be used to create test content, it is better suited to modifying individual pieces of real content for use in teaching and learning. The project partners are currently developing a dedicated test content creation tool to create pieces of test content based on Application Profiles. The tool is being developed to generate correct and incorrect content based on the Application Profile. This automation will remove the need for manually creating content, thereby further reducing the time and cost of conformance testing.

Secondly, conformance testing also highlights the necessity for clear and unambiguous specification documentation. Conformance rules contained in written documentation (as opposed to the specification XSD representation) need to be manually interpreted and entered into the test system for each new specification. Clearly it would be simpler if all requirements could be specified in a machine-readable, unambiguous way. The limitations of XML Schema (Gil & Ratnakar, 2002) have led to the development of a number of alternatives to replace or supplement an XSD file. UML offers the possibility for representing e-learning specifications in an implementation-neutral manner, however this is not yet common practice.

5. Conclusion

In order to achieve the ultimate aim of interoperability between disparate open e-learning systems and content, conformance to base specifications is a vital step. The ability for implementers to create Application Profiles has made e-learning specification conformance testing more complex because of the flexibility it allows implementers. In response, the Telcert project has developed a set of tools which simplify the process of developing Application Profiles and conformance testing against implementations which use such Application Profiles. This article outlined how these tools can be used with established conformance testing techniques to provide a complete solution for testing e-learning specifications. We are currently carrying out tests with a wider range of specifications and exploring the potential for using UML to represent specifications and to derive tests directly from the specification documentation.

6. References

- Binder, R. V. (1999). *Testing Object-Oriented Systems: Models, Patterns, and Tools*, Addison Wesley, Object Technology Series.
- Gil, Y., & Ratnakar, V. (2002). A Comparison of (Semantic) Markup Languages. In Haller, S. M & Simmons, G. (Eds.) *Proceedings of the Fifteenth international Florida Artificial intelligence Research Society Conference*, Florida: AAAI Press, 413-418.
- Hao, R., Lee, D., Sinha, R., & Griffeth, N. (2004). Integrated System Interoperability Testing With Applications to VoIP. *IEEE/ACM Transactions on Networking*, 12 (5), 823-836.
- Kindrick, J., Sauter J., & Matthews R. (1996). Improving Conformance and Interoperability Testing. *StandardView*, 4 (1), 61-68.
- Malek, M. & Dibuz, S. (1998). Pragmatic Method for Interoperability Test Suite Derivation, *Euromicro*, vol. 02, no. 2, 20838.
- ONeill, O., & Nadolski, R. (2005). Implementing E-learning Specifications with Conformance Testing: Profiling for IMS Learning Design. Manuscript submitted for publication.
- Ostrand, T. J., & Balcer, M. J. (1988). The Category-Partition Method for Specifying and Generating Functional Tests. *Communications of the ACM*, 31 (6), 676-686.
- Ramachandran, M. (2003). Testing Software Components Using Boundary Value Analysis. In *29th Euromicro Conference Proceedings*. 94-98. Washington DC: IEEE Computer Society.
- RELOAD (2005). Reusable eLearning Object Authoring & Delivery project website, retrieved 18 August 2005 from <http://www.reload.ac.uk/> .
- Schemaprof (2005). Schemaprof website, retrieved 15 May 2005 from <http://herakles.uni-koblenz.de/schemaprof/> .
- TELCERT (2005). Technology Enhanced Learning Conformance: European Requirements and Testing project website, retrieved 28 July 2005 from <http://www.opengroup.org/telcert/> .

Open Source Flash Projects

The Way Forward For Reusable Learning Objects

Catherine Hurley,
Educational Media Research Centre,
University of Limerick,
Ireland
Catherine.hurley@ul.ie

Timothy Hall
Educational Media Research Centre
University of Limerick,
Ireland
Timothy.Hall@ul.ie

Abstract

Macromedia Flash has always been a popular software product to use in creating multimedia applications for learning. It is particularly suitable for creating reusable learning objects which follow the framework set out in the Cisco Reusable Learning Object (RLO) strategy. This paper will examine how flash is moving towards a more open source development strategy by releasing the source code of its flash player, making its compiled Shockwave Flash file format (.swf) more open. It will also examine how flash developers are becoming increasingly aware of the importance of sharing and reuse. Developers are releasing the source Flash file (*.fla)s of their compiled Shockwave Flash objects under the creative commons licence to facilitate reuse and learning. This will allow learning objects developed in Flash to become more reusable as educational developers can adapt the source code in the original flash file to meet the needs of their students.*

1. Introduction

Macromedia Flash began as a web animation tool favored by artists, but has changed over the years to become a ubiquitous web application platform. The word Flash is used to describe many things. These include the Flash player (virtual machine), Macromedia's Integrated Development Environment called Flash, and compiled Flash bytecode (also known as a Shockwave Flash object or swf). Flash encompasses other technologies such as the ActionScript programming language, RTMP protocol, and AMF data format[1]. For the purpose of this paper, Flash will refer to the Flash platform.

Since the birth of Macromedia Flash years ago, it has become one of the most popular tools for creating e-learning multimedia applications. Its popularity is born from the fact that the Flash IDE is capable of creating rich multimedia content. Macromedia have had huge success with this product, swf has become a well known standard format for multimedia. Although there are other tools available for creating swf type files, so far none have been able to compete with Macromedia's IDE. E-Learning developers have a preference for Flash as an authoring tool particularly because of its in-built learning interactions and its SCORM and AICC compliance with Learning Management Systems.

However, there has always been one major disadvantage of using Macromedia Flash for e-learning development: it is a very expensive tool for cash-strapped education departments to invest in. It has also been closed source in every sense. Developers are bound to using the product and are subject to the company's desires to change the product as they wish. Also, when education departments purchase ready made e-learning courses that have been developed in Flash, they are usually buying the only the pre-compiled swf file. Therefore, they have no way of adapting it without either rehiring the developer or buying the software to adapt it themselves. The source code of the fla file can be compared to the negative to a photograph: it allows reproduction of the original image. Fla files are the source files that can be adapted and recompiled to create a new swf file that meets the needs of the user.

2. Open Source Revolution

Open Source software by definition means that the author has made the source code in addition to the compiled version available. The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software,

the software evolves. People can then improve it, adapt it, and fix bugs.[2] Open Source is a community initiative where developers learn from each others work by studying the source, therefore increasing the knowledge base of the whole community.

The Open source community has developed an Open Source version of most of the popular proprietary software, operating systems such as Linux, applications like Open Office etc have all become popular alternatives. A few attempts have been made to create an open source IDE similar to macromedia Flash but so far none have been as successful. However for the programmers who develop swf through ActionScripting, there have been a few developments.

2.1 New Flash Development Tools

For the artists and animators, the Macromedia IDE will, for the foreseeable future, remain the best way to develop. But for the programmers who develop through Action Scripting, there are several new and interesting open source Integrated Development Environments (IDEs) to use. The combination of the MTASC compiler and Eclipse IDE has turned out to be a very powerful one. MTASC is a powerful open source action script compiler which allows you to develop swf files without the use of Macromedia Flash IDE. The Eclipse IDE has always been a favourite of Java developers, but in this case is used for ActionScript development. ActionScript Development Tool (ASDT) and the Flashout are plug-in s for the Eclipse platform for swf development. The Flashout plug-in provides a similar option to the Test Movie option in Macromedia's IDE [3]. These tools minimise the cost of development as it is unnecessary to buy the license for the Macromedia Flash IDE.

2.2 Open Source Resources

The idealistic values of the open source community have impacted on the Flash development community in the last year. They have begun to see the benefits of sharing resources. Developers have recently started to release the fla file along with the swf under the creative commons license. This represents a new venture for the Flash community. Previously, developers did not want work adapted and reproduced as this minimised profit for the developer. The change of heart is mainly due to the open source revolution that has taken place in the last few years. Macromedia themselves have even taken part in the open source community by releasing their own source code for Flash player. This has made the swf file format more open. They have an application system for requests to see the source code. This enables them to maintain authority concerning who gets to see the code. Mike Chambers of Flash Platform Developer at Macromedia has also developed a simple library that allows the viewer to add a view source menu option to flash content [4].

There are many websites and forums where flash developers are sharing their source files. At the moment, none are dedicated to educational Flash applications. There is a growing culture of sharing resources between Universities presently, with institutions like MIT sharing course materials for many of their courses under their OpenCourseware Initiative. In the future, universities may begin to share their e-learning applications

3. Developing Learning Objects.

There are many definitions of Learning Objects in the e-learning industry but for the purpose of this paper, we will define a Learning object as "the smallest independent structural experience that contains an objective, a learning activity and an assessment" [5]. Each Learning Object should be self-contained but independent of context; that is it should not depend on any other piece of learning content to be complete. This means that the Learning Object can be shared by and reused in multiple lessons or courses.

The Flash platform is suitable for creating e-learning content as it can create a rich multimedia experience for the user. Learning Objects which contain animations, interactions, images and text can be designed which will suit the needs of many different learning styles.

3.1 Reusability of Learning Objects.

The concept of reuse of learning objects has its foundation in the object-oriented approach within software engineering [6]. In any area of software development, reuse of code causes less time and money to be spent on projects, it also allows developers to learn from and improve upon the work that is already in place. This is a big part of the open source community, source

code for projects is posted on the internet and then others take it and improve upon it. This is exactly the type of community work that brought about the Linux operating system.

Open source flash projects will be very beneficial for the reuse of learning objects. Institutions will be able to adapt and re-purpose flash learning objects to suit their own pedagogical needs. This is important to have a culture of this type of sharing in educational environment because often educational institutions do not have the time or the money to develop their own learning objects for e-learning applications.

3.2 Cisco Reusable Learning Object Strategy

Cisco's reusable learning Object strategy is built on the concept of creating small objects that are combined to meet the needs of the learner. The strategy defines a reusable learning object as a collection of seven plus or minus two Reusable information Objects that are grouped together to teach a common job task based on a single learning objective. Reusable information Objects are the building blocks of the reusable Learning Object [7]. Each RIO is defined as a concept, fact, process, principle, or procedure. If an RIO can be equated with an individual component of a learning objective, an RLO is the sum of RIO's needed to fulfill that objective. Each RLO, which also includes introduction, summary, and assessment items, is designed to meet a learning objective derived from a specific job task.

When developing Learning objects using this strategy in a Flash environment, it can be very useful to take advantage of the library feature where you can create mini movies and images that illustrate concept, facts, etc. otherwise known as RIO's . These Reusable Information Objects can be used as many times as needed within the movies. They can also be adapted for a certain section without changing the original in the library as it is only an instance of the original.

If the Learning Object's are created in accordance with this strategy, the learning object becomes truly reusable as down to the smallest chunk can taken out, adapted and reused somewhere else. With the new culture of sharing within the Flash community, this sort of development can be very constructive. When the source file for a Learning Object is released, RIO's from that LO can reuse and re-purposed in other learning objects to suit the needs of the learner.

4. How is Copyright Law affected?

Copyright is a big issue for developers when it comes to sharing their work. Without any kind of licensing laws governing what some one can do with a shared resource, the work could be altered and no credit given to the original author or sold for profit. With original copyright law, all rights are reserved by the user. A license was then developed that allows the author to pick and choose the rights he would like to reserve himself, allowing much more freedom to share but under the author's conditions. Developers who release the source files of their swf can do so without fear their work was being used without credit being given to them.

One option lies in the Creative Commons Licence. This License allows people to share their work with other without fear of theft. It allows people to reserve some of their rights on their work while giving away others under certain conditions. If it is found that the license conditions have been breached in any way, the author has the right to sue. There are eleven licenses to choose from including:

- "Attribution - You let others copy, distribute, display, and perform your copyrighted work and derivative works based upon it - but only if they give you credit.
- Noncommercial - You let others copy, distribute, display, and perform your work - and derivative works based upon it - but for noncommercial purposes only.
- No Derivative Works - You let others copy, distribute, display, and perform only verbatim copies of your work, not derivative works based upon it.
- Share Alike - You allow others to distribute derivative works only under a license identical to the license that governs your work." [8]

Unlike the Free Software foundation's GNU General Public License, Creative Commons licenses are not designed for software, but rather for other kinds of creative works: websites, scholarship, music, film, photography, literature, courseware, etc. There is no direct reference to source code in its license. Creative Commons is a more content license, covering art, audio and video. This is why Flash developers are more likely to use it as their multimedia designs contain these. Creative Commons has "wrapped" some free software/open source licenses with its Commons Deed and metadata if a developer still wishes to use it for something like a Flash project which contains code and multimedia.

5. Conclusion

There is a growing culture of sharing and reuse in the knowledge economy. The Flash community are one of the latest to contribute to its growth through sharing their own work. Even Macromedia who are a proprietary software company have joined in by releasing the source code for their Flash player. Developers have begun to adopt lessons from other open source communities recognising the benefits in sharing their work. The Creative Commons licence has made it more attractive for developers to share their work as they can pick and choose which rights they would like to reserve. With regard to the community of e-learning developers, this is a very positive development. With increased sharing of resources, e-learning developers will be able to reuse more of the content that already exists, saving time and money for their educational institutions.

References

- [1] OSflash (2005) What is Flash?. Retrieved from:
<http://www.osflash.org/doku.php?id=flash&DokuWiki=e9bf25b5c46d5dd95922bb8b8116c660>
- [2] Open Source Initiative explanation. (2005) Retrieved from: <http://www.opensource.org/>
- [3] Rovira, Carlos (2005, June 3) Towards Open source Flash Development. Retrieved from:
<http://www.actionscript.com/index.php/fw/1/towards-open-source-flash-development/>
- [4] Chambers, Mike (2005, April 7) Adding a View Source Menu item to Macromedia Flash Content. Retrieved from:
http://weblogs.macromedia.com/mesh/archives/2005/04/adding_a_view_s.cfm
- [5] Polsani, Pithamber R.(2003, Feb. 19) Use and Abuse of Reusable Learning Objects. Journal of Digital Information, Volume 3 Issue 4, Article No. 164. Retrieved from:
<http://jodi.ecs.soton.ac.uk/Articles/v03/i04/Polsani/?printable=1>
- [6] Littlejohn, Alice (2003) Reusing Online Resources : Chapter 5 :Combining Reusable learning resources and service with pedagogical purposeful units of learning
- [7] Cisco Systems Inc. (2001, Nov). Retrieved from:
http://business.cisco.com/servletwl3/FileDownloader/iqprd/86575/86575_kbns.pdf
- [8] Creative Commons website: Choosing a License. Retrieved from:
<http://creativecommons.org/about/licenses>

e-Portfolios / PDPs: LUSID Developments And Tool Deployment In Bodington

A.C. Marshall¹⁴, S.Grant¹⁵, J. Strivens¹⁶ and S. Mehan¹⁷

Open specifications: IMS: CP, RDCEO, VDEX, E-portfolio; XML; XHTML; Shibboleth.

Open source software: LUSID (PDP/e-portfolio); Bodington (VLE/LMS); Guan Xi (Shibboleth).

Introduction

In the following paper we introduce the concept of Personal Development Planning (PDP) and detail how an open source software project known as LUSID¹⁸ (Liverpool University Student Interactive Database) is able to support this process. We talk about the open standards supported by LUSID and demonstrate how these facilitate integration with other e-learning system, specifically the Bodington¹⁹ Virtual Learning Environment (VLE) / Learning Management System (LMS).

Personal development planning

There is a requirement for all UK Higher Education institutions to offer their students the opportunity to undertake PDP by the 2005/6 academic year. PDP is defined by the Quality Assurance Agency for HE²⁰ as *a structured and supported process undertaken by an individual to reflect upon their own learning, performance and / or achievement and to plan for their personal, educational, and career development*. In many cases this process will be supported by an electronic tool, however, this is not a requirement and a paper based solution would be acceptable.

QAA envisage that PDP will help students: become more effective, independent and confident self-directed learners; understand how they are learning and relate their learning to a wider context; improve their general skills for study and career management; articulate their personal goals and evaluate progress towards their achievement; and encourage a positive attitude to learning throughout life²¹.

In order to address this need, in 1997 the University of Liverpool began to develop a system known as LUSID. After a number of years of closed development at Liverpool, the JISC²²-funded SPWS²³ project ensured that the software was open sourced and it is now available from SourceForge²⁴ under the Lesser GPL²⁵.

What is LUSID?

LUSID is a **fully customisable** web-based PDP tool; as it supports PDP, it can be classified as an e-Portfolio. It acts as a complement to a Virtual Learning Environment such as Blackboard, WebCT or Bodington providing facilities that none of these systems offer.

¹⁴ OUCS, University of Oxford adam.marshall@computing-services.oxford.ac.uk

¹⁵ Information Strategists, Inst.co.uk

¹⁶ Centre for Lifelong Learning, University of Liverpool

¹⁷ Library and Information Systems, UHI Millennium Institute

¹⁸ <http://www.lusid.org.uk/>

¹⁹ <http://www.bodington.org/>

²⁰ <http://www.qaa.ac.uk/>

²¹ <http://www.qaa.ac.uk/academicinfrastructure/progressFiles/guidelines/progfile2001.asp>

²² Joint Information Systems Council (UK) <http://www.jisc.ac.uk/>

²³ Skills Profiling Web Service <http://www.elframework.org/projects/spws/>

²⁴ <http://lusid.sourceforge.net/>

²⁵ <http://www.gnu.org/copyleft/lesser.html>

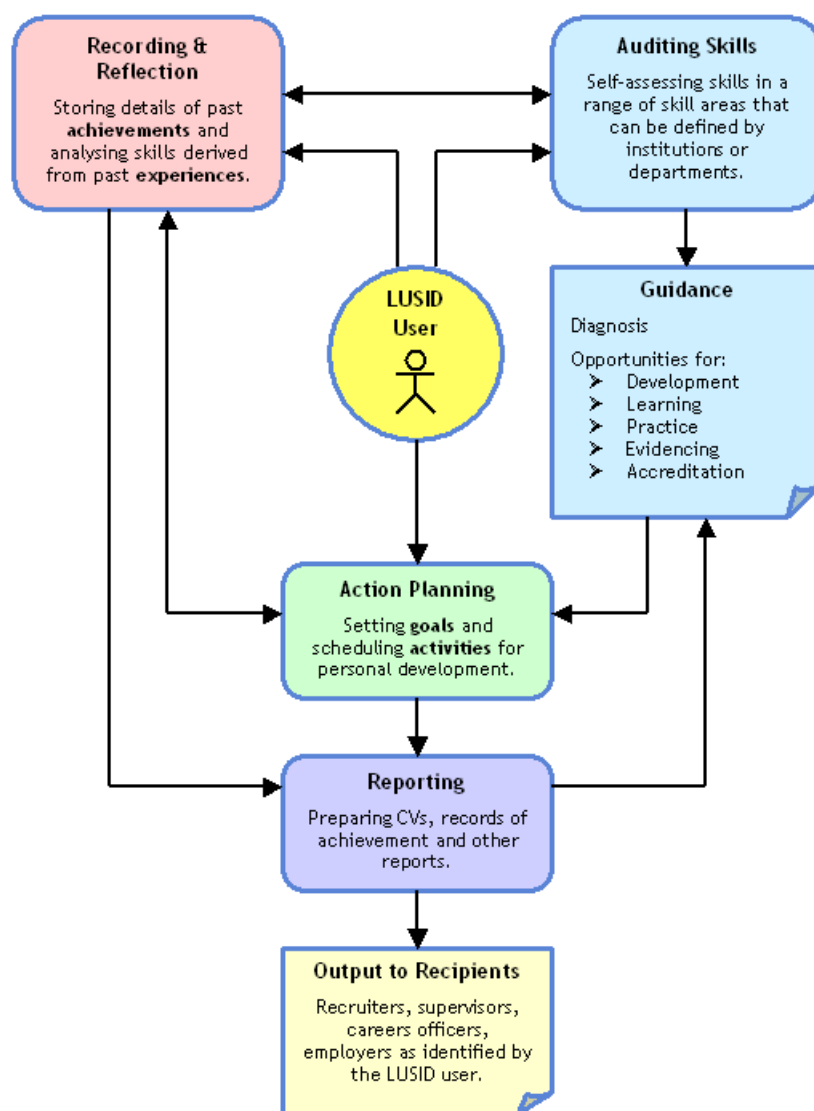


Figure 1 Schematic representation of LUSID

LUSID offers support in four main areas: recording and reflection, action planning, auditing skills and reporting (see Figure 1). Guidance and support is provided throughout the system to promote independent personal development planning.

The recording section can be used to collate details of learning experiences including employment information, educational achievements and (work-based) learning logs. All experiences can be analysed in terms of skills used and knowledge gained. The action planning section allows the planning of goals and activities, either by the use of an interactive Gantt chart, or by importing a predefined plan. An individual's skills can be audited and, based on this, guidance provided to help improve competence. The reporting section can be used to draw together a selection made by the user from all data stored within LUSID in the form of a CV (or equivalent report) which can then be sent via e-mail to interested parties (and edited in a word processor if so desired). Alternatively records can be sent (with the user's knowledge) to other data storage systems where they can be accessed by anyone with the necessary permissions.

All pages in LUSID (written in XML²⁶) can be customised to include local terminology, resources and guidance. An individual department or tutor group can have its own customised set of

²⁶ An extension of XHTML

pages which, for example, will allow specific exercises to be set, a subset of skills to be audited, or predetermined action plans for course elements to be imported.

LUSID is implemented as a Java servlet and is able to run on a totally free platform: Linux, Postgres and Tomcat.

In addition to being an interactive system, LUSID has a number of web service interfaces. It is able to import and export personal data in IMS LIP format²⁷, expose its skills framework in IMS RDCEO²⁸ and IMS VDEX²⁹ format and act as an anonymous PDP web service (see later).

Behind the scenes

There are a few innovative concepts underpinning LUSID that have ensured its success over the years. In general these concepts have mapped well to subsequent developments by bodies such as the IMS Global Learning Consortium. LUSID has also informed the development of the UKLeaP (BS 8788)³⁰ and IMS e-portfolio³¹.

Skills framework

From its very inception, LUSID has supported a pluggable and extensible skills framework.

Ideally, skills developed in one context should contribute to the evidence of potential competence in a different context. One can too easily imagine an employer being frustrated that a graduate or school leaver presents them with little evidence of the skills they need for employment. This may be because the relevant skills are simply not acquired; but increasingly it may be that they cover similar ground, but are represented in different ways. Both the individual skills represented, and the overall structure, may differ between different "skills frameworks".

We identified the idea of *shared skill concepts*^{32 33}. These are *non-overlapping* chunks of skill. In an IT context you may think of word processing (WP) skills as a shared concept; database skills would be another example.

Each skill concept may be further decomposed (for example, formatting text); this applies recursively, (for example, emboldening text, italicising text, and so on,) see

Figure 2. The 'shared' nature can be illustrated as follows: italicising a word is the same in MS Word as it is in MS Front Page; once you have the skill to do this in Word, you also are able to do the same thing in Front Page! (This is obviously a very simple example for illustrative purposes only.)

For use with an e-learning system, these shared skill concepts may be grouped into what one may term 'skill areas'. Skill sets are collections of one or more skill areas.

²⁷ Learner information package: <http://imglobal.org/profiles/>

²⁸ Reusable definition of competency or educational objective:
<http://www.imglobal.org/competancies/>

²⁹ UK format for PDP data: <http://www.imglobal.org/vdex/index.html>

³⁰ <http://www.bsi-global.com/index.xalter>

³¹ <http://www.imglobal.org/ep/>

³² Implementation of a Portable Skills Framework, Grant, S., Marshall, A. C., and Strivens, J., Recent Research Developments in Learning Technologies vol III (2005) pp940-945 III International Conference on multimedia & ICT's in Education. ISBN 609-5997-X

³³ A skills meta framework for UK education, Grant, S., see:
<http://www.elframework.org/projects/spws/SPWS-meta-framework-final.pdf/view> (a long and detailed document)

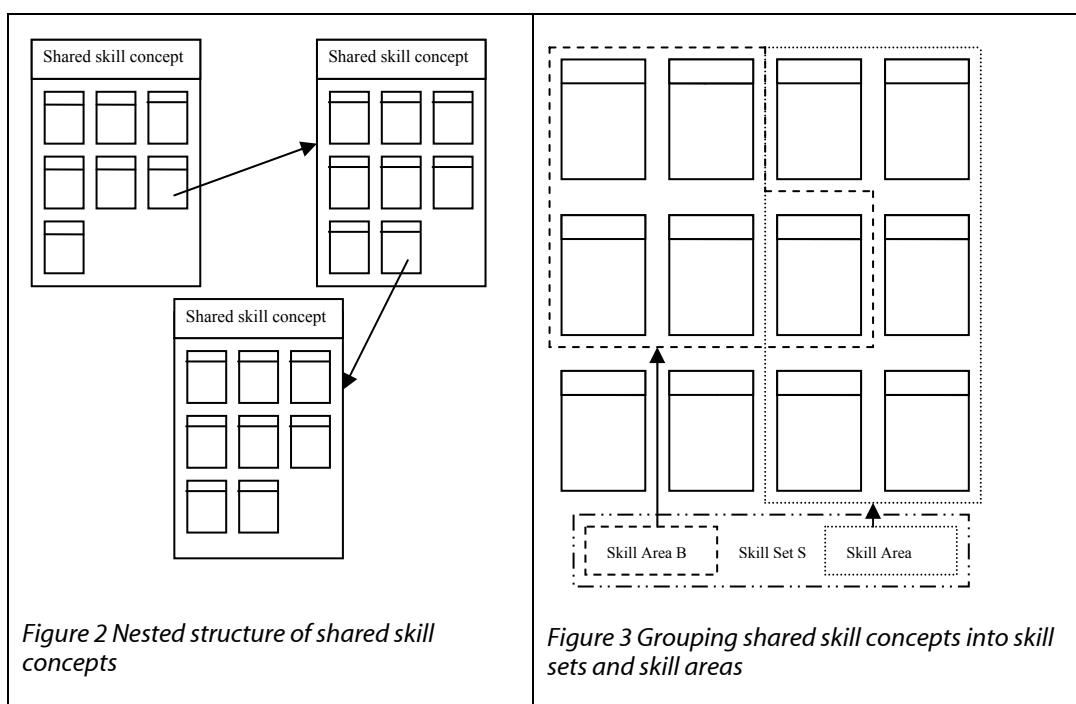


Figure 2 Nested structure of shared skill concepts

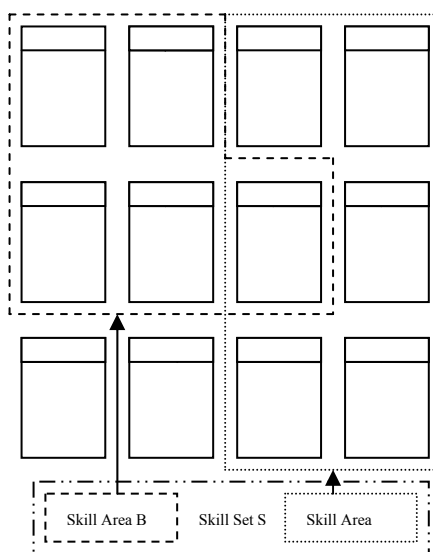


Figure 3 Grouping shared skill concepts into skill sets and skill areas

Figure 3 shows how a number of skill concepts can be grouped into skill areas; note how a shared skill concept may be a member of more than one skill area. The skill areas can be grouped into skill sets. One valid skill, S, set may comprise skill areas A and B. Another skill set may comprise skill areas A, C and D. As all skill areas use same base framework, they are able to be related to each other. (Skills concepts, areas and sets are all represented by IMS RDCEO and VDEX.)

Skill concepts can be gathered together into skill areas by an institution or department as they see fit – the concepts are fixed throughout the domain but the ways they are combined into skills areas / skill sets are not. This will allow, for example, Medical Schools to have their own personalised definitions of skill areas tailored to the local way of thinking - but as all skills areas are drawn from the same framework, all schools would be able refer back to the same base set of definitions.

The SPWS project added a RESTian web service interface to LUSID to allow other e-learning systems to interrogate LUSID and obtain a definition of its skills framework in terms of IMS RDCEO³⁴. Further information can be found via www.lusid.org.uk.

Personal data

The structure which LUSID uses to represent personal data is very similar to that employed by IMS LIP (and latterly IMS e-portfolio).

IMS LIP and LUSID both represent activities, goals, qualifications, certificates and licences, competencies, interests and so on. As IMS LIP was developed independently from LUSID, the team were greatly reassured to discover that the two initiatives achieved more or less the same conclusions regarding data structure.

Mapping LUSID's data structures to IMS LIP is relatively straightforward, moreover, experience with LUSID³⁵ has directly led to enhancements to UKLeaP (the UK 'version' of IMS LIP) and to

³⁴ For those who delight in reading such things (!), LUSID's framework is available as a zip file from: <http://www.elframework.org/projects/spws/skills-with-skillsets-and-medical-rdceos.zip/view>

the new IMS ePortfolio. LUSID places heavy emphasis on reflections. (Reflective learning³⁶ is highly thought of in the UK and is central to LUSID's model.) Use of reflection in LUSID led directly to the inclusion of the 'reflexion' element in the aforementioned specifications.

Reflection

Reflection is supported by LUSID in a number of ways. A series of pages can be linked to form a review/planning process, taking the user through a set of leading questions about their thoughts, feelings and experiences. The questions are designed to help the user to analyse their experiences and direct their attention to particular aspects of it. Particularly useful is the facility to re-present to the user a record made at a previous point, to prompt comparison, awareness of change and development or simply to remember earlier thoughts. Here LUSID differs from more conventional e-portfolio systems by using a clearly-designed set of activities to generate records. The activities are all recognisably "PDP activities"³⁷.

The skills framework can be used in two ways to support reflection. A user recording an activity/experience can be prompted to analyse what skills were used, and to note when such an activity provides a particularly good piece of evidence for the skill in question. As more activities are undertaken and reflected upon, the learner builds up a portfolio of demonstrable uses of skills which of course can be exported as part of the learner record. This is extremely useful when compiling a CV or other report which needs to demonstrate a specific set of skills requested by an employer.

The same skills framework can be used to audit the user's general level of confidence. The use of the skills framework in these two ways means that the system has the potential to offer guidance to the user by comparing records. Where a user is confident in a skill but has no evidence recorded of activities demonstrating that skill, or where a user has recorded low confidence but much experience, the system can 'point this out', allowing further opportunity for reflection!

LUSID, the JISC e-framework and web services

The JISC e-framework³⁸ (formerly ELF³⁹) is an international effort to develop a service-orientated approach to the development and integration of computer systems in the sphere of learning, research and education administration. Three of the boxes (services) within the Learning Domain are relevant to LUSID: competency, e-portfolio and personal development.

LUSID is able to be used as a component in three different ways; it can act as the supplier of a competency framework, as a 'stand-alone' recording system or as a client of a user agent such as the open source Bodington VLE/LMS (or indeed any other system that is able to handle IMS Content Packages⁴⁰ comprising XHTML and IMS QTI v2⁴¹).

The first use has been discussed above. It is envisaged that Shibboleth⁴² would be used to integrate LUSID with another e-learning system. Indeed, Bodington (v2.4 and upwards) includes the Guan Xi⁴³ Shibboleth Identity Provider which means that the user and group store in Bodington can be used to pass account information between systems. It is planned for LUSID to (optionally) use the Guan Xi Service Provider to control access. This work will be undertaken in 2006.

The third use shows LUSID acting as a service which can be remotely consumed. As part of the SPWS project, both Bodington and LUSID were enhanced to allow LUSID to offer an anonymous RESTian⁴⁴ skills profiling service and for Bodington to consume this service.

³⁵ MLEs for Lifelong Learning support program, <http://www.cetis.ac.uk/members/llsp>

³⁶ http://www.heacademy.ac.uk/resources.asp?process=full_record§ion=generic&id=69

³⁷ <http://www.cetis.ac.uk/members/PDPcontent/viewActivityTypes>

³⁸ <http://www.e-framework.org/>

³⁹ <http://www.elframework.org/>

⁴⁰ <http://www.imsglobal.org/content/packaging/>

⁴¹ Question and Test Interoperability: <http://www.imsglobal.org/question/>

⁴² <http://shibboleth.internet2.edu/>

⁴³ <http://guanxi.sourceforge.net/>

⁴⁴ <http://webservices.xml.com/pub/a/ws/2002/02/20/rest.html>

LUSID is able to generate an IMS Content Package (CP) which can be used as a 'journal' or 'logbook' to reflect upon an activity. The structure and content of the CP is defined by a series of XML templates within LUSID. These templates are driven by both the skills framework and by LUSID's internal data structure. Anything that can be represented within the LUSID system can also be exported as a CP.

The web service takes two parameters, the skill set and the set of pages to be used. So, for example, there is a page-set that defines a straightforward skills audit, "How confident are you at skill X?" "How confident are you at skill Y?" The same page-set can be used to audit medical skills or language skills or business skills dependent on what skill set is supplied as an argument.

Likewise, the page-set may define different reflection activities. As well as a skills audit there is also a page-set that allows one to document and reflect upon a work-based learning activity. The same skill set may be supplied to different page sets so that learners will have a consistent view of the skills that should be within their horizon. It is quite a simple process to create many more templates and skill sets which could be used in a multitude of different learning activities.

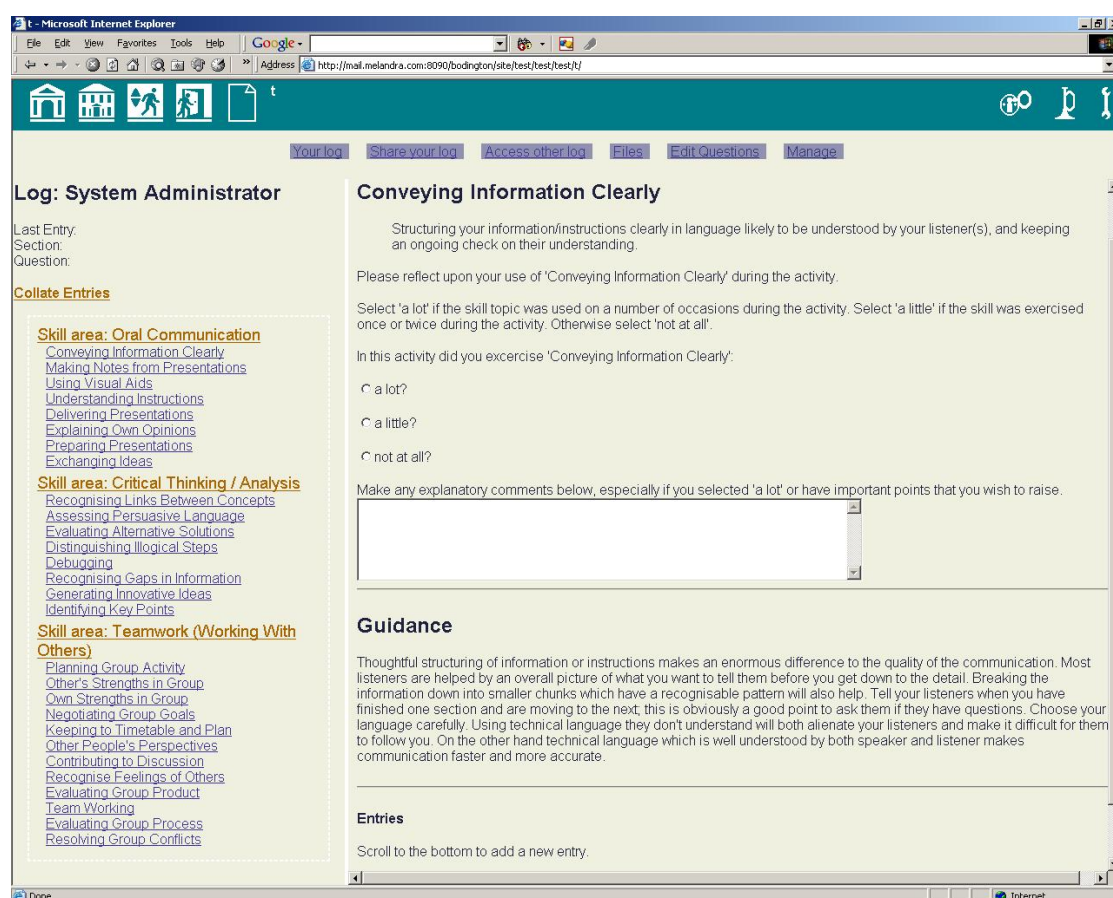


Figure 4 Screen shot of a Bodington logbook rendering a CP produced by LUSID

Bodington has a logbook tool. Under normal circumstances a tutor creates a series of section and questions which individual learners fill in. The logbook acts like a (very) basic e-portfolio with learners able to collate selected responses in a format suitable for printing. This logbook tool has been enhanced to allow it to talk to LUSID. Bodington receives a CP and then renders the CP within the logbook framework by using the APIS web service⁴⁵ (see Figure 4).

Currently Bodington must store any data entered by the student but one day it is envisaged that the data will be stored in a PDP system or e-portfolio. (There are sizable issues surrounding

⁴⁵ <http://www.jisc.ac.uk/index.cfm?name=apis>

the storage of distributed personal data and its impact on PDP. The WS4RL website contains a fuller discussion of this topic⁴⁶.)

The above facilities will be available in Bodington 2.8 (due for release at the end of 2005) and LUISD 2.2 (due early 2006).

Summary

We have introduced three open source e-learning software systems and shown how they are able to work together. The decision by Oxford University to embrace open source software and open standards has had a very positive effect on the gamut of e-learning opportunities that can be offered to our students. With easy access to the source code and increasing knowledge of the systems we have been able to build our own customised learning environment and offer an excellent value for money to the University.

There are currently plans afoot to use more open source software within the Bodington framework. We plan to use the CETIS IMS Enterprise Services⁴⁷ toolkit to allow access Bodington's person, group and membership store by other components of the e-framework and to use MVNForum⁴⁸ as an alternative discussion forum. We will also integrate an (as yet undecided) open source Wiki.

⁴⁶ <http://www.elframework.org/projects/ws4rl/>

⁴⁷ <http://www.elearning.ac.uk/features/entws>

⁴⁸ <http://www.mvnforum.com/mvnforumweb/index.jsp>

Practical pedagogical uses of IMS Learning Design's Level B

Daniel Burgos and Rob Koper
Educational Technology Expertise Centre (OTEC)
Open University of the Netherlands
Valkenburgerweg 177; PO Box 2960
6401 DL Heerlen; The Netherlands
{daniel.burgos;rob.koper}@ou.nl

Abstract

One of the main concerns while making lesson plans in IMS Learning Design is how to model practical, pedagogical, actual scenarios in IMS Learning Design and how IMS Learning Design can help to move real lesson plans, fully focused on pedagogical and didactical uses, to an open e-learning specification without getting lost in the process within technical issues. So, is it possible to make it and how?

This paper intends to put together the pedagogical requests of teachers and learning designers and the technical approach needed to realise them using the Level B of IMS Learning Design. Through different examples and specific uses we describe both the pedagogical needs and the suggested coding and we link them to provide a joint view that allows to point out a discussion formula where didactical end-users needs for teaching meet a pedagogically expressive specification.

Keywords

IMS Learning Design, Unit of Learning, Level B, Adaptive Learning, Collaborative Learning, Assessment, Instructional Design

1. Introduction

IMS Learning Design, or IMS LD (IMS, 2003), is a new specification that jumped into the e-learning panorama to build a bridge from pedagogical face-to-face actual models to e-learning and blended learning frameworks. IMS LD allows the modelling of regular lesson plans of teachers in units of learning to be run in a certain online platform. In the last year a number of tools (engines, players and editors) have been born, like CopperCore (Vogten and Martens, 2005), CopperAuthor (Van der Vegt, 2005), Reload (Bolton, 2004), Sled (OUUK, 2005) and several others and improved in parallel with efforts at dissemination, like the European UNFOLD Project (UNFOLD, 2004) and Learning Network for Learning Design (OUNL, 2004). In addition, efforts have been made on interoperability between open source tools based on IMS LD or related to it, like Moodle (Dougiamas, 2004).

The Open University of The Netherlands (OUNL) works intensively providing engines, tools, examples and extensive documentation about the specification, related specifications, and feasible applications of them. In 2004 and 2005, OUNL has also carried out the task of disseminating IMS LD, funded by UNFOLD, together with several European universities. In 2005 a number of face-to-face meetings and online activities have been carried out by these two institutions and some strong useful feedback has been gathered from end-users, mainly from teachers and learning designers. One of the main concerns is how to move from an actual lesson plan in a classroom to a well-structured Unit of Learning in IMS LD without losing pedagogical expressiveness. The second concern is how to create these units in an easy way for non-technical end-users.

2. Basic structure of IMS Learning Design

IMS Learning Design is divided into three levels. Level A, with the definition of the method, plays, acts, roles, role-parts, learning activities, support activities and environments is the core of the specification, and contains the description of the elements that configure IMS LD and the coordination between them. For instance, role-parts define what activities must be taken by a role in order to complete an act and, subsequently, a play.

Level B, adds properties, conditions, monitoring services and global elements to Level A, and provides specific means to create more complex structures and learning experiences. Properties can be used as variables, local or global, storing and retrieving information for a single user, a group or even for all the characters involved. Through these mechanisms the learning flow can be changed at the run time, as decisions can be made taking into account dynamic content. Logically it is the level to express most of the pedagogical needs concerning adaptation, personalization, feedback, tracking and several other usual requests of teachers and learning designers.

Lastly, Level C adds notifications to Level B, meaning an email sent and a show/hide command to a specific activity, depending on the completion of another one.

3. Practical pedagogical uses

Level B provides several facilities concerning properties, conditions and the monitoring service, as we have already explained above, and all of them can be used in a wide range of applications. The most frequent uses are described from a didactical perspective: active learning, collaborative learning, adaptive learning, personalization, dynamic feedback, runtime tracking, ePortfolios and new forms of assessment. Although the source code provided shows properly the dataflow and the information structure it is not the original code in IMD LD notation. Some superfluous details and additional tags have been ruled out in order to get a neater explanation. In addition, the engine CopperCore (op. cit.) has been used to run and try all the examples, representing the only current tool capable of a successful running of all the levels in the specification.

3.1 Active and collaborative learning

Collaborative learning means (Cole and Engestrom, 1993) to share information on a peer basis, student-student and student-teacher, chasing to consolidate knowledge, criticize opinions and remarks, provide some new ideas in the light of others' work or insert new topics for discussion and a collective debate, for instance. This means that the same specific information can be seen for different persons and that a constant data interchange flow comes out. Also, it means that each member of the group can use it matching their personal goals inside this group or inside the course.

A second option allows a teacher to monitor the progress of his students, analysing dynamically all the contributions coming from the course and providing a proper feedback to them in both ways, one by one and collectively. This way, a bilateral information flow between learner and tutor is established, aimed at academic and personalized good use. Level B provides the component 'monitor' and allows the viewing of self properties and properties from the others in a structured way. These properties must be defined previous its use, and initialised if numeric data types are managed, and can be operated into formulas, as further is shown. Following are two examples about definition and initialisation:

```
<loc-property identifier="LP-LA-1-completed">  
  <title>Response to the initial quiz</title>  
  <datatype="boolean"/>  
  <initial-value>true</initial-value>  
</loc-property>
```

The property 'LP-LA-1-completed', with the sentence 'Response to the initial quiz' as a title or label, is a Boolean type and its initial value is set to TRUE. This variable is useful to store the current state of a learning activity to know whether it is finished.

```
<locpers-property identifier="LP-personalgoals">  
  <title>Which are your goals for this course?</title>
```

```
<datatype="text"/>
</locpers-property>
```

The property 'LP-personalgoals', with the title 'Which are your personal goals for this course?', is a Text type and it is not initialised so far. This variable is used to store personal information from the user and, therefore, there is no previous content. It is a local property and depends on every user.

If we want to use this last property ('LP-personalgoals') it can be read and saved along the normal course flow:

```
<Id:set-property ref="LP-personalgoals"/>
```

This code line allows to write ('set') some content in the property.

```
<Id:view-property ref="LP-personalgoals" property-of="self"/>
```

This code line allows to read ('view') the internal value of our own property ('self').

Moreover, this property can also be traced with the 'monitor' component. For instance, the following code line allows the reading ('view') of the property of a different student ('supported-person')

```
<Id:view-property property-of="supported-person" ref="LP-personalgoals"/>
```

To start this monitoring action, firstly the component 'monitor' must be set-up inside an environment (in this specific case):

```
<environment identifier="E-personalgoals">
  <title>Which are the goals of the others?</title>
  <service identifier="S-personalgoals">
    <monitor>
      <role-ref ref="Student"/>
      <title>Goals of the other students</title>
      <item identifierref="R-personalgoals"/>
    </monitor>
  </service>
</environment>
```

Above, the 'monitor' service is defined for a learner ('Student'). This means that every student can view the content of his other classroom partners' properties. In case a tutor needs to view students' properties a similar structure can be written, providing a proper tracking of each participant in a course.

3.2 Adaptive learning and personalization

In order to illustrate this section we take the example 'Learning to listen to Jazz' (Tattersall and Burgos, 2005) initially developed in EML and later adapted to IMS LD. In this Unit of Learning a student can follow a course about Jazz and can choose two different itineraries, thematic and historic, based on his preferences. Also, some actions of monitoring can be tackled in the way described previously.

Adaptive learning (Shuell, 1988) pretends to choose, collect and show some contents to a student, coming from a common data base, and depending on the student's initial profile and on the progressive results gained during the running of the course. Personalization complements adaptive learning, providing the capacity to choose specific features for each student, like content, look and feel, assessment and itinerary, all together inside a pre-made collective learning framework.

Regarding the concept of adaptive learning, and taking the referenced example, Jazz, a property called 'LP-choose-itinerary' is set-up to know whether the user has chosen one of the two itineraries. Each of them is described inside their Activity Structure, 'AS-historic' and 'AS-thematic', previously defined in the manifest and out of the scope of this paper. All the process to choose an itinerary is programmed as a flow of conditions, taking one option or the other depending on the value of this property. Both Activity Structures are hidden in the beginning, when any value is inside the property yet:

```

<conditions>
<if>
  <no-value>
    <property-ref ref="LP-choose-itinerary"/>
  </no-value>
</if>
<then>
  <hide>
    <activity-structure-ref ref="AS-thematic"/>
    <activity-structure-ref ref="AS-historic"/>
  </hide>
</then>

```

The student can take one of the two options, 'historic' and 'thematic', available in a combo box. If the user takes the option 'thematic', his related structure 'AS-thematic' is shown and the non-related structure 'AS-historic' is hidden. The same things happens the other way around:

```

<if>
  <is>
    <property-ref ref="LP-choose-itinerary"/>
    <property-value>thematic</property-value>
  </is>
</if>
<then>
  <show>
    <activity-structure-ref ref="AS-thematic"/>
  </show>
  <hide>
    <activity-structure-ref ref="AS-historic"/>
  </hide>
</then>

```

These two different structures are able to provide non-identical contents or the same ones re-organized in several ways dealing with two complementary or opposite approaches and all is managed inside the same manifest coming with the Unit of Learning.

On the complementary concept, personalization, a simple case is to get the personal details of a student in the course. In the previous section we showed how to define and fill a property. Another possibility is to group several properties under a single name to operate easier. For instance

- a) defining a property, String type, and a second one, Integer type, and initialising this last one to zero

```

<locpers-property identifier="LP-name">
  <title>your name</title>
  <datatype datatype="string"/>
</locpers-property>
<locpers-property identifier="LP-age">
  <title>age</title>
  <datatype datatype="integer"/>
  <initial-value>0</initial-value>
</locpers-property>

```

- b) grouping both properties

```

<property-group identifier="LP-group-personalinfo">
  <title>personal details</title>
  <property-ref ref="LP-name"/>
  <property-ref ref="LP-age"/>
</property-group>

```

- c) requesting the related information to the user ('set') and showing the results ('view'), on demand

```
<ld:set-property-group ref="LP-group-personalinfo" property-of="self"/>
<ld:view-property-group ref=" LP-group-personalinfo" property-of="self"/>
```

3.3 Dynamic feedback and runtime tracking

In order to illustrate this section we take the example 'GeoQuiz' (Burgos, 2005). In this Unit of Learning a student answers a questionnaire consisting of five questions and related additional feedback is given depending on the chosen responses. Later, a numeric final valuation is provided, meaning an average and a final remark based on it.

As a specific application on adaptive learning, IMS LD also allows to work with a) dynamic feedback, providing contents adapted to the students progress in a certain Learning Activity, and b) reading the results depending on his activity.

Regarding a) the provision of contents depending on a certain student evolution we have already seen the possibility in the previous example of making a call inside the manifest file itself to different structures with different, similar or equal contents and uneven orders. All in all, we are talking of showing and hiding an activity or a structure, while being related to resources, pointing themselves to external files.

Another possibility though, deals with using classes to modify the visibility of specific content inside a external file pointed by a resource, this is, outside the manifest file. These classes are XHTML layers defined inside XML files besides the imsmanifest.xml file.

Although the mechanism that allows the hiding and showing of these layers/classes looks like that already explained for Activity Structures, this action is now carried out in a file outside the manifest and, therefore, it runs rather differently:

```
<if>
  <is>
    <property-ref ref="Answer1"/>
    <property-value>C</property-value>
  </is>
</if>
<then>
  <hide>
    < class="Feedback_Wrong"/>
  </hide>
  <show>
    <class="Feedback_Right"/>
  </show>
</then>
```

In this previous example we show the class 'Feedback_Right' only if the content of the answer 'Answer1' is equal to 'C' (this response is picking it up from an enumerated list), and hide the class 'Feedback_Wrong', with a very different content. In the external file, the value of the property 'Answer1' is set and both are defined, classes and their content:

- a) an answer is chosen ('set')

```
<p>Your answer is:
<set-property ref="Answer1" of="self"/></p>
```

- b) the classes are defined

```
<div class="Answer1_Wrong">
  <p><view-property ref="Answer1"/> is not right</p>
</div>
<div class="Answer1_Right">
  <p>Congratulations!</p>
</div>
```

Regarding b) the reading and interpretation of results depending on the user activity, we can make arithmetic calculations with the stored values along the course and provide a contextual feedback based on them. If we envisage two questions with two answers and two feasible values for each answer, 0 and 100, we could:

a) define every property

```
<locpers-property identifier="QuestionTrue1">
  <datatype datatype="integer"/>
  <initial-value>0</initial-value>
</locpers-property>
```

b) assign a value based on a specific answer

```
<if>
  <is>
    <property-ref ref="Answer1"/>
    <property-value>C</property-value>
  </is>
</if>
<then>
  <change-property-value>
    <property-ref ref="QuestionTrue1"/>
    <property-value>100</property-value>
  </change-property-value>
</then>
<else>
  <change-property-value>
    <property-ref ref="QuestionTrue1"/>
    <property-value>0</property-value>
  </change-property-value>
</else>
```

c) and, lastly, calculate a simple average with the two answers as arguments

```
<change-property-value>
  <property-ref ref="sum"/>
  <property-value>
    <calculate>
      <divide>
        <sum>
          <property-ref ref="QuestionTrue1"/>
          <property-ref ref="QuestionTrue2"/>
        </sum>
        <property-value>2</property-value>
      </divide>
    </calculate>
  </property-value>
</change-property-value>
```

3.4 ePortfolios and new forms of assessment

Traditional assessment is grounded in the confrontation between the user knowledge against a machine or against the teacher's knowledge. But this is not the only one way. New approaches to learning and teaching must develop new forms of assessment. Evaluation should be given throughout the learning flow and not just as an isolated resource a) checking the level of knowledge of a student before to decide his best itinerary, b) checking whether a concept is properly understood before stepping further with the next one and c) providing high quality feedback to maintain high motivation. Also d) scenarios consisting of several students must take into account the sharing of individual responses from the group for a common valuation and a collective debate (Koper and Tattersall, 2005). To formalize any of these proposals in LD is not difficult but is hard work, using the notes we have drawn through these pages. These four possibilities can be implemented as a combination of conditions and properties as long as formalizing evaluation questionnaires is feasible with IMS LD. Another challenge is too combine IMS LD and IMS Question and Test Interoperability Specification (IMS,

2003a). Although the implementation of QTI is outside the scope of this article, we can suggest how to integrate both specifications, an issue that also concerns LD. Basically, the sequence is the one already commented on and it describes the relationship between an environment and a linked resource. The only difference is the type of the reference (this will be a QTI type, like 'imsqti') that will link to a file written to this specification ('question_1.xml'):

```
<environment identifier="Env-1">
  <title>A test linking QTI and IMS LD</title>
  <learning-object identifier="LO-QTI-question-1">
    <title>First question</title>
    <item identifier="I-1" identifierref="R-que-1"/>
  </learning-object>
</environment>

<resource identifier="R-que-1" type="imsqti">
  <file href="question_1.xml"/>
</resource>
```

Lastly, the use of ePortfolios is feasible if the original drive of the IMS ePortfolio specification is held (IMS, 2002). It means to save and share data externally to any application and keep it standardized inside the information package, making possible that the information is consumed and managed in different places and systems. Global properties in IMS LD allow the definition of private information of a user (globpers-property), of a group (property-group) or, working as a constant, for everyone (glob-property). The first of these properties is also called 'portfolio-property' because it allows the incorporation of files to the Unit of Learning in runtime. As usual, this use needs two different moments:

- a) to create and set-up the global property

```
<globpers-property identifier="GP-suggestions">
  <global-definition uri="GP-suggestions">
    <title>Suggestions about the course</title>
    <datatype datatype="file"/>
  </global-definition>
</globpers-property>
```

- b) to use the property in a XML file

```
<div class="upload-file">
  <p>Choose the file with your suggestions:</p>
  <ld:set-property ref="GP-suggestions"/>
</div>
```

4. Conclusion

Although Level A in IMS LD provides the base of any Unit of Learning and draws the skeleton of any learning flow, Level B provides several mechanisms to improve and make lesson plans stronger. Properties, conditions, global elements and monitoring services are the four main features of the specification that make Level B the most powerful one. This means that the learning scenarios could include important practical pedagogical uses and be modelled in IMS LD, mainly using the Level B layer. Active learning, collaborative learning, adaptive learning, personalization, dynamic feedback, runtime tracking, ePortfolios and new ways of assessment are all key teaching and learning resources that point out how to model in IMS LD. Bearing this in mind, it is fair to say that in the current state in the development of tools, it is not a trivial issue to relate the pedagogical approach to a running Unit of Learning. Many of the current tools are technically built but not pedagogically gifted. Users still need to know some of the specification to build Level A units and quite a lot of XML coding to create Levels B and C units. In practice, it means that nowadays is not so easy for a non-technical end-user to edit and run Units of Learning in actual lesson plans.

The most recent discussions on IMS LD tools suggest the design of a top layer more focused on usability, pedagogical and navigational issues than on technical issues. Improving the message and the metaphor in the editor and in the player will certainly enrich the view of the teachers and lesson plan builders and make their work easier and faster. This way, teachers and learning designers could build Units of Learning bottom-up (focused on their experience and needs) instead of top down (focused on understanding the specification itself), using an inductive approach.

5. References

1. Bolton (2004) Reload Project. United Kingdom: The University of Bolton, The University of Strathclyde and JISC. Available at <http://www.reload.ac.uk/editor.html>. Retrieved August 31st, 2005
2. Burgos, Daniel (2005) Geo Quiz Example Unit of Learning. Available at <http://moodle.learningnetworks.org>. Retrieved on September 29th, 2005
3. Cole, M., Engestrom, Y. (1993) A cultural-historical approach to distributed cognition, In G. Salomon (Ed.), *Distributed cognitions*. New York: Cambridge University Press
4. Dougiamas, M. (2004) Moodle. Available at <http://moodle.org/>. Retrieved on August 31st, 2005
5. IMS (2002) ePortfolio. Boston: USA. Available at www.imsglobal.org. Retrieved on July 29th, 2005
6. IMS (2003) Learning Design. Boston: USA. Available at www.imsglobal.org. Retrieved on July 30th, 2005
7. IMS (2003) Question and Test Interoperability. Boston: USA. Available at www.imsglobal.org. Retrieved on July 29th, 2005
8. Koper, R., Tattersall, C. (2005) *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Germany: Springer Verlag
9. OUNL (2004). *Learning Network for Learning Design*. Heerlen: Open University of The Netherlands, OTEC. Available at <http://moodle.learningnetworks.org>. Retrieved on August 14th, 2005
10. OUUK (2005) Sled player. Open University of United Kingdom. Available at <http://sled.open.ac.uk> Retrieved on September 29th, 2005
11. Shuell, Th.J. (1988) The role of the student in learning from instruction, *Contemporary Educational Psychology*, 13, 276-295
12. Tattersall, Colin and Burgos, Daniel (2005) Learning to Listen to Jazz Example Unit of Learning. Available at <http://moodle.learningnetworks.org>. Retrieved on September 29th, 2005
13. UNFOLD (2004) UNFOLD Project. Available at www.unfold-project.net Retrieved August 30th, 2005
14. Van der Vegt, Wim (2005) CopperAuthor. Heerlen: Open University of The Netherlands. Available at www.coppercore.org. Retrieved on July 29th, 2005
15. Vogten, H., Martens, H. (2005) CopperCore 2.2.4. Heerlen: Open University of The Netherlands. Retrieved at www.coppercore.org. Retrieved on October 9th, 2005

Metadata and Application Profiles: a Data Model

Robert Schuwer¹, Henk Hangyi², Darco Jansen¹
Open University of the Netherlands, Heerlen¹; Mmatch, Rotterdam²
robert.schuwer@ou.nl , hangyi@xs4all.nl , darco.jansen@ou.nl

Abstract

In the Ruud de Moor Centre of the Open University of the Netherlands (RdMC), a lot of learning materials is developed to support new teachers and their educators. Reuse of material and packaging for different environments makes the task of labelling products with metadata elements important. Based on a world wide standard, we have developed application profiles of metadata elements. In this paper the background of the RdMC application profile is described. The core elements of a data model for an application to support labelling according to an application profile are presented. This datamodel offers flexibility in defining and linking application profiles to usergroups. Based on this data model, a metadata editor, YAME, is developed.

1. Introduction

The Ruud de Moor Centre (RdMC) of the Open University of the Netherlands (OUNL) is supporting a typical category of real life learners: career switchers who enter a teaching job in a school (in this paper we will address these learners as “new teachers”). While working as a teacher they have to acquire their formal qualification in one or two years, for which the new teacher, the school and the teacher training institute enter a tri-partite contract. This on the job training, i.e. in the school, is becoming increasingly important in the solution of the problems caused by the shortage of teachers, especially in primary and secondary education.

At this moment (October 2005) the RdMC carries out over 25 projects. Several disciplines and practitioners in the field are involved. Products to be developed are, amongst others, knowledge bases, communities of practice for distant coaching and several instruments for (self)assessment.

Materials, developed in a project, can be reused by other projects in the RdMC. For example several projects can use the same video, each in its own context. Also within projects materials can be used in several ways (e.g. by creating variants of the product for different target groups). The users (new teachers and their coaches) demand web based delivery of material tailored to their specific needs (just-in-time, just-for-me and just-enough) (Dekeyser et al (2004)).

This kind of tailoring requires a flexible, multi-purpose environment for learning and learner support, accessible by all actors according to their needs, preferences and contexts, from the workplace as well as from the distance (i.e. by IT-tools and/or consultancy) (Stijnen (2003)). The RdMC has chosen for an open, modular architecture for such an environment that allows for different authoring systems, a flexible repository and delivery to a variety of learning/working environments by a wide range of media (web, dvd / cd-rom, paper, mobile devices). It implies that the RdMC will not develop yet another LMS, but will support the common authoring systems and delivery environments used in the educational field. The kernel of this environment will be the repository (i.e. a number of interrelated repositories).

One of the key factors to create, maintain and use such well structured, interrelated repositories is a metadata model. By making agreements about the metadata elements (which attributes will we use? How do we describe these attributes? Which vocabulary will be needed?) the products will be exchangeable and findable. These agreements are articulated in an RdMC application profile.

In this paper we will elaborate on the metadata model and the RdMC application profile. First we will present the application profile and its backgrounds. The process of labelling products with metadata will be described then. Both the application profile and the process of labelling determine the datamodel for storage and retrieval of metadata. The presentation of this datamodel is the kernel of this paper. Based on this datamodel a metadata editor is developed. We will present this metadata editor.

Although these developments took place in a learning environment, the principle that has led to the datamodel and, therefore, the datamodel itself, can be generalised to and used in other domains than the educational domain.

2. What is an application profile?

For learning materials several standards for metadata do exist. Examples of such standards are Dublin Core (Anonymous (2003)), SCORM (Sharable Content Object Reference Model, see reference), IMS (see reference) and LOM (Sloep et al (2004)). As was already sketched in the introduction, the RdMC will support the common authoring systems and delivery environments used in the educational field. Therefore, adhering to a standard is important for the RdMC. In the Netherlands, LOM is developing to a de facto standard. We therefore decided to start with the LOM standard when we started thinking about the set of metadata elements for the RdMC.

LOM has developed into an IEEE-standard. It contains both objective and subjective metadata elements. Objective metadata elements are product characteristics that are independent of the content, the user or its use. Examples are an ID, file size and copyrights. Subjective metadata-elements are product characteristics that describe its content, a user or its intended use. Examples are the title, key words and user judgements.

LOM consists of more than 70 metadata elements. This gives the advantage that detailed descriptions of products can be made. Its big disadvantage is that labelling a product with all these metadata elements is very time consuming. Therefore, LOM is not suited to be used as-is.

To cope with this situation, more and more user communities are specifying application profiles (Jansen et al (2005)). An application profile is a set of schemas which consist of data elements drawn from one or more namespaces, combined together by implementers, and optimised for a particular local application (Heery and Patel (2000)). In our application profile, LOM fulfilled the role of the namespace as is mentioned in the definition. An application profile in its simplest form defines the metadata elements to be used.

We have defined an overall RdMC application profile. The metadata elements in this profile fall into three categories:

- Category 1: consists of the metadata elements that are mandatory when labelling a product. Examples are the title of the product, the intended user for the product (e.g. a new teacher in primary education) and the intended usage of the product (e.g. a case study).
- Category 2: consists of the metadata elements each product should be labelled with, but where assignment of the value is taken care of by the metadata editor (when the product on hand is electronically available). Examples of such metadata elements are file type, file size and date of creation.
- Category 3: consists of the metadata elements that are not mandatory when labelling a product. Examples are level of aggregation of the product (asset, paragraph, module.), estimated time for using the product and description of the product.

Apart from naming the metadata elements and declaring them mandatory or optional, in an application profile this can be extended by defining the values each metadata element can take and describing dependencies between metadata elements. In the RdMC application profile, three types of dependencies exist:

- Type 1: using a metadata element when labelling a product is dependent of the value of another metadata element for that product. An example is the situation when labelling a knowledge base. The value for the metadata element Type of product ("knowledge base") excludes the metadata element Filetype for this product.

- Type 2: characteristics of a metadata element are dependant on the value of one or more other metadata elements. An example of this type of dependency is given by the metadata elements Producttype and Runtime. For a video (Producttype is "Video") Runtime is a mandatory metadata element.
- Type 3: The list of values for one or more metadata elements is dependant on the value of another metadata element. An example of this type of dependency is given by the metadata elements Type of school and Content domain. The values for Content domain differ for primary and secondary education.

Each project has to use the RdMC application profile when labelling a product with metadata elements. A project however has the possibility to shape the RdMC application profile to its own needs. The following changes are allowed:

- Not using metadata elements that fall into category 3 of the RdMC application profile. These metadata elements are not offered to the author when labelling a product.
- Adding new metadata elements
- Adding values to a list of values of a metadata element
- Not using values in a list of values of a metadata element
- Renaming metadata elements ("dialects")
- Declaring metadata elements that are optional in the RdMC application profile mandatory when labelling a product.

When shaping the "minimal" RdMC application profile, the following advantages can be gained:

- Within a project, specific terminology can be used. This is an advantage for project members who want to reuse the materials. The end user will not be confronted with project specific terminology, because the terminology is recognized as synonym of common terminology,
- The amount of work when labelling a product with metadata elements can be reduced by offering only the essential metadata elements and lists of values.

The main advantage, however, is the possibility to adapt the application profile to the needs of specific user groups. The RdMC application profile is targeted to an organisational unit (RdMC), but the project specific application profiles are targeted to specific project groups. A project specific application profile mimics the needs of the users of the results of the project, even if those needs deviate from what is common in the field.

The characteristics of an application profile are one source of demands for system functions that will support labelling of products with metadata elements. Another source of demands is in the process of labelling. This process will be described in the next chapter.

3. The process of labelling with metadata

After the definition of an application profile, there are two types of activities that have to be performed for labelling products according to the application profile:

- Setting up the environment
- The actual process of labelling

Both types of activities will be described in more detail.

3.1. Setting up the environment

As was mentioned in chapter 2, within an organisation several application profiles can exist. Each application profile has to be implemented in an information system to be available for the actual process of labelling. A system administrator or application owner typically does these activities. Availability of an application profile means:

- Metadata elements that are part of the application profile should be registered in the information system.
- For each metadata element, its characteristics within the application profile should be available.
- The list of values that are used by the metadata elements in the application profile should be defined.

- Dependencies between the metadata elements should be implemented. It should be taken care of that each product is labelled in accordance with the dependencies. Typically, there are two ways this can be established. One way is that the user dependency is taken care of before the user actually fills in the value for a metadata element when labelling a product. The other way is checking on the dependency after filling in the value. The first way is preferred because of the efficiency of the process for the user.
- Application profiles have to be linked to user groups. This way, the right application profile can be retrieved when a user has to label a product with metadata elements.

When the environment is set up, the state of the environment can be compared to a 'factory on Sunday afternoon'. All equipment is available, ready to use, stock is filled with parts, but there are no employees present.

3.2. The actual process of labelling

A user of the metadata application logs on to the application when he has to label a specific product with metadata elements. After login, the user is known to the system and a list of application profiles is presented to him. From this list, the user selects an application profile. The user is presented a list of metadata elements. For each element, a value has to be submitted, either by typing in the value or by selecting one or more values from a list of values. When dependencies exist between metadata elements, the result of submitting a value to one metadata element can be that the list of values of another metadata element will be changed or even a metadata element can be made non-selectable. The order in which the metadata elements are presented to the user therefore is dynamically determined by the application. After submitting all the values, eventually some checks on conformance to the application profile are done by the application. When all is correct, the data is written to the database and the user can label another product.

It is not required that the product to be labelled is available electronically. For these types of product the metadata elements of category 2 do not exist.

4. Data Model

The following figure shows the logical datastructure for the core entity types of the metadata application. A rectangle represents an entity type, whereas an arrow represents a n:1 relationship between two entity types (the "1" being on the arrow side).

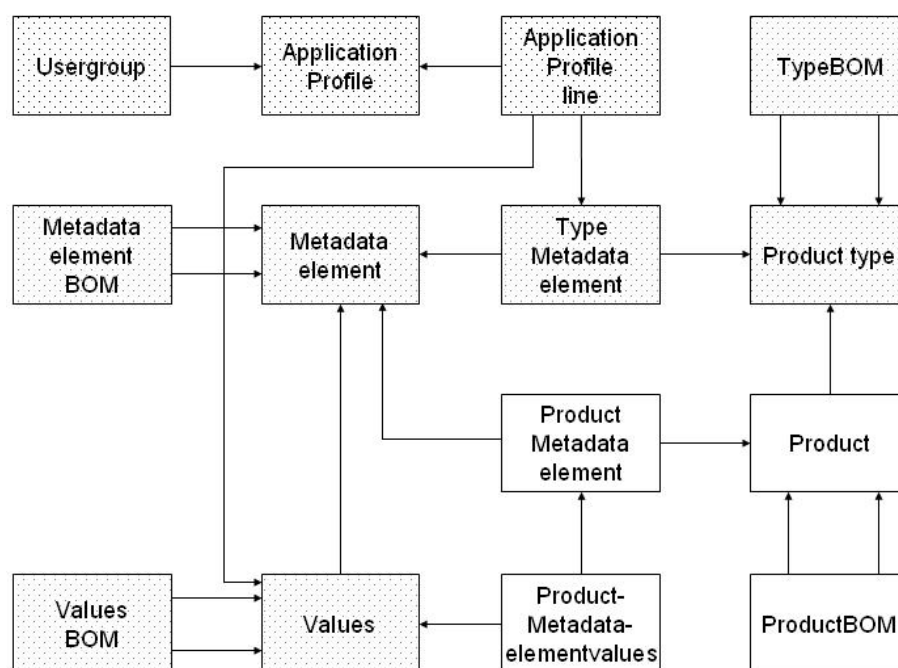


Figure 1. Logical datastructure for a metadata application based on application profiles

The entity types are divided into two categories. The punctuated entity types represent the data that was described in chapter 3.1 (“the factory on Sunday afternoon”). The non-punctuated entity types represent the actual labeling of products. In table 1 a description of each entity type is given.

Table 1. Description of the entity types

Entity type	Description
Product type	Type of product (e.g. knowledge base, case)
Metadata element	Label used for describing a product
Values	Values a metadata element can take
Type Metadata element	Describes which metadata elements can be used when labelling a product of the given product type.
Application profile	Container for the prescriptions that counts when labelling a product
Usergroup	Describes the user community and links it to an application profile
Application profile line	Describes a value a given metadata element can take when labelling a product of a given product type
Type BOM	Describes the structure of a product type (“Bill Of Material”)
Metadata element BOM	Describes the structure of a set of metadata elements
Values BOM	Describes the structure for lists of values
Product	Describes an actual product that has to be labelled
Product metadata element	Describes the metadata elements for which a value is given during the labelling process
Product metadata element values	Describes the value given to a metadata element for a given product
Product BOM	Describes the structure of the product

As can be seen from the descriptions of the entity types, the datastructure is not restricted to application profiles in the field of learning. Also, it can be seen that application profiles can be defined for all kinds of usergroups (ranging from organisations to specific groups of users).

5. Application: metadata editor

Bottom line, there are users who have the tedious task of labelling a great amount of products with metadata elements. To make this task as easy as possible, an application is needed that supports the process as is described in chapter 3.2 and that is based on the data model, presented in chapter 4. This kind of application is called a metadata editor.

Based on the data model of chapter 4, a metadata editor YAME (Yet another Metadata Editor) is developed. The editor is built in Java, using the MMBase open source environment (Becking (2005)). One of the existing applications built on the MMBase platform is the Electronic Learning Environment Didactor. Functionality in this application could be reused by YAME. Some additions where necessary:

- Didactor can not implement multi-level valuelists (the Values-BOM of the datamodel).
- Some constraints on the data model for YAME could not be enforced by the data model of Didactor
- The three types of dependency and its influence on the way it becomes visible for the user were not supported by Didactor

YAME can set automatically the following metadata elements of category 2: creation date, date of last modification, file format, file size, player, and playtime.

Another implementation of a metadata editor based on the data model of chapter 4 is done in a closed environment (the author system Content-e (see reference)).

Taylorling the editor to the application profile that fits the user at most is one of the measurements we have taken to accomplish less resistance for labelling products. Another important measurement is the process of defining the application profile. Thereby, we have

strived for maximum user involvement. The near future will learn us if we have reached the goal: a great number of labelled products, targeted to specific user groups.

6. Conclusions

Application profiles are necessary when standards are used while labelling products with metadata elements. For making these application profiles available, an information system is necessary that supports the process of creating and maintaining application profiles on the one hand and supports the user when he labels products using application profiles. A data model for this kind of information systems is presented. This datamodel is not restricted for use in the field of learning materials alone, but it can be used in all domains where application profiles are used. Based on this datamodel, a metadata editor is presented.

The first version of the metadata editor is released now. Next, we will gather user experiences in using this editor. Undoubtedly, these experiences will lead to improvements of the metadata editor and more insight in advantages and disadvantages of using application profiles.

7. References

Becking, J. et al (2005), MMBase, an open source content management system. IBM Systems Journal, Vol. 44 (2).

Dekeyser, H., Rijn, H. F. van, Jansen, D., (2004), Teacher Training on the Job: a Generic Metadata Modelling Approach for Personalised Learning and Learner Support, in: "Information and Communication Technology and Real-life learning, New education for the knowledge society". Springer Science, Business Media. In preparation.

Jansen, D., Schuwer, R., Dekeyser, H. (2005), RdMC-applicatieprofiel, een poldermodel in metadata. Working paper Open University of the Netherlands - Ruud de Moorcentrum, Heerlen. (In Dutch). Online available: http://www.ou.nl/Docs/Expertise/RdMC/Paper_applicatieprofiel_eeen_poldermodel_in_metadata.pdf

Sloep, P.B., Benneker, F. Gorissen, P. (2004), Conceptnorm voor leerobject-metadata, Dutch translation of the Draft Standard for Learning Object Metadata, IEEE P1484.21/D6.4, NEN, Delft. (In Dutch). Online available: <http://www.nen.nl/nl/act/spec/leertech/>

Stijnen, P.J.J. (2003), Leraar worden: 'under construction'?. Inaugural lecture. Maastricht. (In Dutch)

WWW Metadata and Metadata Editors:

Dublin Core Metadata Initiative (DCMI): <http://dublincore.org/>

Dublin Core Metadata Editor: <http://www.ukoln.ac.uk/metadata/dcdot/>

IMS Global Learning Consortium: <http://www.imsproject.org/> and

http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html

Reggie, The Metadata Editor: <http://www.metadata.net/dstc/>

Imse Vimse LOM Metadata Editor: <http://projekte.learninglab.uni-hannover.de/pub/bscw.cgi/0/5890>

MMBase <http://www.mmbase.org>

Author system Content-e: <http://eng.content-e.nl>

WWW Application Profiles:

Heery, R. and Patel, M. (2000): Application profiles: mixing and matching metadata schemas.

<http://www.ariadne.ac.uk/issue25/app-profiles/>

Monash University Library: <http://www.lib.monash.edu.au/metadata/standards.html>

How a FIRM (Flexibility, Innovation, Robustness, and Maturity) Argument for FOSS (Free and Open Source Software) Can Displace FUD (Fear, Uncertainty, and Doubt).

Derek Briton
Athabasca University
derek.briton@athabascau.ca

Abstract

This paper discusses how a coalition of Athabasca University (AU) faculty successfully promoted Moodle, an open source learning management system (LMS), as a viable alternative to two major proprietary LMSs: WebCT Vista and Lotus Notes. The evaluation tool a group of core users developed to determine AU's choice of LMS is described and the evaluation results are touched upon. The evaluation process, however, was not a neat, technical exercise, but rather a process of debate, contention, disagreement, and compromise. Because an LMS resides at the confluence of the social and technological, choosing an LMS is not a purely technological act, but rather a communicative process that can be fraught with political, economic, and cultural factors, as well as personas. Advocates of open source software need to remain fully cognizant of this fact and be prepared to calmly provide evidence of flexibility, innovation, robustness, and maturity (FIRM) whenever institutional and/or personal objections appeal to and/or promulgate arguments against open source software based on fear, uncertainty, and doubt (FUD).

1. Introduction

Athabasca University (AU) faculty first learned of the decision to adopt WebCT Vista as the University's Learning Management System (LMS) in a University-wide email circulated by Athabasca's Chief Information officer (CIO) on Wednesday, September 17, 2004:

In order that we can provide stable, sustainable and world class courses and learning experiences for students, I have recommended, after extensive research and review, that AU adopt WebCT Vista as its learning management system (LMS). I have further recommended that we migrate to this platform through a transition process with an objective of completion by the end of 2006.

This seemingly arbitrary decision deeply troubled AU's faculty for a number of reasons; the most significant, however, were: a lack of consultation; the declaration that choosing an LMS was a non-academic matter; and the absence of factual evidence to support the choice of WebCT Vista.

This precipitated a faculty-led intervention that pressured the administration into agreeing to a transparent evaluation of 3 enterprise-level LMSs, at least one of which would be open source. Since there are a number of open source LMSs, those advocating an open source alternative had to carefully assess the most viable open source candidate. The overwhelming choice was Moodle, primarily because of its proven track record, its extensive feature set and modular architecture, and the vibrancy of its development community.

2. Selection of an Evaluation Team

Before the evaluation could begin, representatives from across the University had to be recruited and constituted into an Evaluation Group. These evaluators were drawn from program administration, student registration and record keeping, system administration, course development and delivery, Web development, help desk and student services, and central computing systems (security, database management, support). The evaluators were then divided into those whose interests were primarily administrative, and those whose

interests were primarily hands-on: a Core Evaluation Team of 12, and a hands-on Testing Team of 20-plus. (Of the 20-plus Test Team, 17 would submit comprehensive evaluations).

3. Evaluation Process

WebCT Vista, Lotus Notes, and Moodle were identified as the three contenders. WebCT Vista was the first LMS to be evaluated, and it was out of the process of inquiry and testing that ensued that an initial listing of evaluation criteria (needs assessment) emerged. The WebCT Vista evaluation also provided evaluators with a much greater knowledge of the complexities involved in the course production, delivery, and administration processes, as well as the centralized computing services infrastructure required to support and maintain those services. The Evaluation Group met 9 times (not including demonstrations and training sessions) between October 2004 and May 2005. The scope of early meetings ranged widely, and discussions were often intense, and it was in these initial meetings that fear, uncertainty, and doubt (FUD) about open source software first emerged. Rather than confront these arguments on a philosophical or political-economic basis (democracy vs. corporatism, licensing fees vs. free and open access, etc.), which could very easily have served to polarize the evaluators into camps of "for" and "against" open source software, efforts were focussed on emphasizing the flexibility, innovation, robustness, and maturity (FIRM) of the open source candidate (Moodle). Economics, often the single-most important factor in the choice of an LMS, especially when resources are scarce, did not play a major role in this case: the necessary resources were available to implement whichever LMS proved most appropriate for AU's needs. As the evaluation process continued, there were many opportunities for members of the Evaluation Group to communicate via email and face-to-face regarding the flexibility, innovation, robustness, and maturity of Moodle over WebCT Vista and Lotus Notes. Fortunately, there were several examples of corporations abandoning proprietary software in favour of open source in the media. For example, in June of 2005, a Business Week Online article entitled "The Power of Us" explores how mass collaboration is affecting business, and notes:

Nowhere has that phenomenon happened faster than in software. Collaborative open-source development is rapidly moving beyond basic utility software like Linux to mainstream applications as well. An especially eye-opening example is SugarCRM Inc., which provides an open-source version of customer-relationship management software now dominated by Siebel Systems (SEBL) and salesforce.com Inc. (CRM)

This article served to substantiate the claims of a June 2004 article published in Forbes, entitled "Cheapware," wherein the corporate migration from proprietary to open source software is explored:

Craig Murphy has had enough. As chief technology officer at Sabre Holdings, which runs the world's largest airfare and ticketing network, Murphy has spent millions of dollars on database and other software from companies like Oracle. But last year, when Sabre was building a new computer system for online shoppers, Murphy took a flyer on a database program from a little-known company in Sweden that charges only \$495 per server computer, versus a \$160,000 list price for Oracle. Guess what? The Swedish stuff works great. Fired up, Murphy is hunting for other places to use the cheaper software, called MySQL. "We're just not going to pay license fees for those databases like we used to. We'll download free stuff off the Internet before we do that," Murphy says. "I believe this is the future of computing."

Interestingly, MySQL's incursion into the database domain is mirrored by that of other open source software in the Web server, operating system domains: Apache and Linux. In a paper entitled "Will Open Source Software Become An Important Institutional Strategy in Higher Education?," an executive briefing paper delivered to the Alliance for Higher Education Competitiveness in May of 2005, Rob Abel notes Apache's growing popularity Web-wide, and the increasing popularity of open source software targeted at the educational market:

Apache has over three times the market share of its nearest competitor (Microsoft) according to a poll by Netcraft of publicly available web sites – and Apache's share appears to be growing. As for specific higher education focused products, there is one, uPortal, which appears to have made a significant penetration in the higher education enterprise. On its web site uPortal notes 79 institutions that have deployed uPortal and 68 in the process of implementation.

Gavin Clark, in "Developers Mad for It," published in The Register, in October of 2005, notes:

MySQL is fast approaching majority market share among software developers, with 44 per cent using the open source database to meet their needs. Use of MySQL has surged 25 per cent during the last six months according to EDC.

And according to an August, 2005 world-wide survey of the server market conducted by IDC:

Linux servers posted their 12th consecutive quarter of double-digit growth, with year-over-year revenue growth of 45.1% and unit shipments up 32.1%. Customers continue to expand the role of Linux servers into an ever increasing array of workloads in both the commercial and technical segments of the market.

Another example of the growing acceptance and popularity of open source software offered in the Forbes article is that of E-Trade:

E-Trade, the online brokerage firm, has slashed its IT budget by 50% through measures that include replacing Sun Microsystems hardware and software with Intel-based computers running Linux and the open-source Apache Web software. Now E-Trade is considering dumping Web programs made by BEA Systems and replacing them with an open-source alternative. Though E-Trade still uses commercial database programs from IBM, Oracle and Sybase, "We would look at alternatives," says Joshua Levine, chief technology officer at E-Trade, in New York.

It was by circulating and discussing evidence such as this that the Evaluation Group was alerted to the advantages/strengths of open source software. Such discussions served to dispel the fear, uncertainty, and doubt regarding the viability of open source software, which cleared the way for the Testing Team to focus on evaluating the strength's and advantages of proven open source software—flexibility, innovation, robustness, and maturity, and control:

Fundamentally, the difference between open source and proprietary software has to do with control. Where the open source license imparts freedoms to use, modify, and redistribute the software, the proprietary license restricts use, modification, distribution, and more. Vendors of proprietary software restrict access to and use of the source code because the source is knowledge—and knowledge is power. (Coppola & Neeley, p. 3, 2004).

Eventually, after much discussion and debate, the Evaluation Group's focus narrowed and consensus began to emerge as the testing sessions progressed. Between meetings, exchanges continued via email, and a Wiki was eventually implemented to facilitate dialogue and track progress.

Once the Moodle evaluation was undertaken in earnest, evaluators soon discovered its modular structure renders it exceedingly flexible; it's freely available source code supports pedagogical innovation and creativity; and it's ongoing development by an international coterie of users, and support of cohorts of up to 91, 000 learners (The Open Polytechnic of New Zealand, <http://campus.openpolytechnic.ac.nz/moodle/>), were clear evidence of its maturity and robustness.

4. The Rating System

The LMS evaluation tool evolved throughout the testing phase and the debates that ensued, and was finalized in May, 2005. Evaluation criteria were grouped under 5 main headings: mandate, systems administration, cost, instructional design, and teaching and learning. The first (mandate) identifies concerns directly related to the University's unique mission and mandate as an open, distance-education university: monthly course registrations and extendable completion dates, individualized and cohort-based learning, affordability, accessibility (for visually, physically impaired), and connectivity (for those located in remote geographic locations). The second (systems administration) identifies concerns related to integration with existing systems (registration, authentication, library, etc.), security, and standards compliance (SCORM, XML, etc.). The third (cost) identifies concerns related to licensing fees, hardware/software, integration with existing systems, and support and in-house training. The fourth (instructional design) identifies concerns related to learning objects, the

separation of content from delivery, and user friendliness. The fifth (teaching and learning) and final heading identifies a number of concerns related to the teaching and learning experience.

5. Methodology

Data was collected in the form an Excel spreadsheet that assigned a weight and priority rating to each criterion and automatically tabulated the result. Each evaluator's weighting and priority score was averaged to ensure consistency. The weighted scores were then totalled to provide a score for each LMS. The scope of this paper precludes the possibility of discussing the evaluation tool and process in detail, but both will be presented in greater detail when the paper is delivered at the conference.

6. References

Hof, Robert, D. (2005). The power of Us: Mass collaboration on the Internet is shaking up business. Business Week Online. June 20, 2005.
http://www.businessweek.com/magazine/content/05_25/b3938601.htm

Clarke, Gavin. (2005). Developers mad for it. The Register, October 18, 2005.
http://www.theregister.co.uk/2005/10/18/mysql_marketshare_numbers/

IDC. World-wide survey of the server market conducted, August, 2005.
<http://www.idc.com/getdoc.jsp?containerId=prUS00223005>

Lyons, Daniel. (2004). Cheapware. Forbes.
<http://www.keepmedia.com/ShowItemDetails.do?itemID=536336&extID=10030>

Coppola, Chris, and Neelley, Ed. (2004). Open source – opens learning. Why open source makes sense for education. <http://www.rsmart.com/assets/OpenSourceOpensLearningJuly2004.pdf>

Abel, Ron. (2005). Will open source software become an important institutional strategy in higher education? http://www.ahec.org/research/indepth_articles/open_source0505/open_source0505_toc.html

Overcoming Proprietary Hurdles: CPOINT as Invasive Editor

Andrea Kohlhase
Digital Media at Education (DiMeB), University of Bremen
kohlhase@informatik.uni-bremen.de

Abstract

Even though open standards and tools are the ones to aim for, many powerful excellent proprietary tools are used in practice. Once a user is accustomed to a special software tool it is not easy to convince her of a different equivalent open source one. One way to overcome this obstacle is to build open source software for proprietary applications ("invading" them). CPOINT is a semantic, invasive editor for Microsoft PowerPoint (PPT). It enables the user to annotate objects on a slide by providing a user interface to the semantic (open source) XML data standard OMDOC from within PPT. The semantic annotations enable the user to manage PPT slides, their content can also be made available to all other OMDOC applications e.g. the learning environment ActiveMath.

1. Introduction

In this paper, we want to contrast scientific expectations with users' realities concerning educational software. I will argue that the educational potential of digital media is lost if users are forced to learn new tools for basic, already mastered functionalities (like text editing) before being able to use a new functionality (like semantic markup editing). The trade-off between the potential and the investment into achieving it has to be acknowledged and subsequently tipped towards the latter.

This specifically concerns open software for education as the educational community usually is rather reluctant to integrate new technology into their standard work-flows. Insofar as digital media are already used, they become a hurdle to tipping the trade-off scale in the wanted direction. Since open software is frequently at the edge of new technology and the user's previously learned software is typically standard software (like MS Word), we can identify these specific preoccupations as "proprietary hurdles". In order to overcome them, we will develop the idea of integrating "what's already known to the user" into open source projects. In particular, we argue that there are many basic functionalities (like editing, calculating, organizing) that in principle are independent of newly developed technologies. We only have to find a way to invade the already known interfaces so that they can be extended by the new functionality.

As an example of such *invasive technology* we look at the semantic editor CPOINT that invades MS PowerPoint (PPT) to enable the user to capture, share, and reuse content from within PPT.

2. The User Riddle

We want to exemplify the "User Riddle", i.e. the contrast between scientific expectations and users' realities concerning educational software, with a short description of the history of the Semantic Web. In 1989, WWW inventor and W3C director Tim Berners-Lee started to envision the gigantic open source project "Semantic Web". His road map is based on his understanding of the Semantic Web as "a Web in which machine reasoning will be ubiquitous and devastatingly powerful" [Berners-Lee, 1998] assuming that its data exists in machine-processable documents. Even though he was aware of the fact that "instead of asking machines to understand people's language, it involves asking people to make the extra effort" [sic], he was so impressed by the possibility and necessity to manage the Web's data (see also [Berners-Lee & Fischetti, 1999], or [Berners-Lee et al., 2001]), that he oversaw the cost-benefits ratio involved for its users.

Even though the idea of a "Semantic Web" established itself beyond scientific communities, it hasn't come to real life yet. Some, for instance Tim Bray (who e.g. co-authored XML and recently served on the W3C Technical Architecture Group), argue that the Semantic Web is still missing a "killer app" which would trigger other applications and user acceptance. In particular, he invited the community in 2003 to participate at a challenge concerning RDF, a vital component of the Semantic Web [Bray, 2003]. Interestingly, he put up a rather long (more than three years) period for this challenge, either hoping to attract real killers (in terms of applications) or accounting for the difficult motivational requirements on users. In October 2004, Berners-Lee was asked in an interview why people don't seem too excited about the Semantic Web [Frauenfelder, 2004]. Acknowledging this lift-off problem, he explained it with a still-to-come paradigm-shift from small scale to big scale in the users' heads. The underlying question why users haven't taken up the call circumscribes the "User Riddle". We will now address the peculiar discrepancy between scientists' and users' expectations concerning Information and Communications Technologies (ICT), especially educational aspects.

2.1. Scientific Expectations

Non-surprisingly, the value of ICT seems almost self-evident for scientists who developed it. The technology comprises the values, beliefs, and hopes of its designers. More surprisingly, ICT was not only taken up by other scientists but by the society as a whole (see e.g. the European Research Program [ERA, 2005]).

On the one hand, the potential of ICT is generally seen to consist in its distribution capacity, in its networking options, in its elaborate communication possibilities: in short, in its broadening of a user's action range. On the other hand, ICT enables global availability of information, huge data storage capacity, and personal data publication: in short, it empowers a user's knowledge construction system. It is rather tempting to conclude that ICT is just great for users and that this should be motivation enough to pick it up. Sure enough: the success of the WWW outstripped every expectation and user acceptance of the WWW is very high.

Scientists often affirmatively assume an organic unity between a user's potential advantages resulting from ICT applications and her attitude towards the technology [Schluss, 2002]. Moreover, attitude is too often assumed to translate directly into actual behavior, i.e. action. Therefore, it is scientifically expected that users wholeheartedly embrace ICT in theory and in practice.

In particular, this was true for expectations concerning ICT and education, especially eLearning. Here, the enticing potential of digital media in general together with the ICT infrastructure under an educational perspective is mainly made up of its built-in mutability (supporting individual learner/teacher preferences), time and space independence (lending itself to new learning slots), and global information source (offering incredibly many learning materials). Even though this potential is accepted throughout the educational community (learners as well as teachers), it is another instance of the User Riddle, i.e. it hasn't borne as much fruit as expected in analogy to the Semantic Web example.

2.2. Users' Realities

Let us now investigate why the transformation from theory to practice doesn't work out. In contrast to [Kohlhase & Kohlhase, 2004a], where we likened the users' realities to the prisoner's dilemma, here I want to take into account a more pedagogical and design perspective. If an action is considered to be a subjective, intentional activity, then the execution of actions are dependent on the recognition of the value of their effects. This recognition is guided by the following questions [Heid, 2004, pp. 146]:

- Who influences the value of an action's outcome?
- Whose interests are implemented in that value?
- Who assigns responsibility?

In order to take a more detailed picture, we take Learning Objects in the context of the educational Semantic Web as an example (see e.g. [Anderson & Whitelock, 2004]).

The intrinsic value of learning objects consists basically in their reusability by their authors as well as other users. Therefore, the economic factor "educational rationalization" is supposed to become the main motivation esp. for teachers to author learning objects in the short run - to be awarded in an uncertain future by making use of "all" available learning objects. We see that the user itself can at best influence her action's outcome indirectly by her participation in a self-organizing process. This motivation is rather second hand as she has to trust in the scientific expectations. It is clear that there are scientific and institutional interests in

educational rationalization, but it isn't obvious that the user will gain personal advantages if she executes the action. Finally, we address the question of responsibility assignment. Here, the set-up is relevant. Generally, open source systems will assign the responsibility to the (virtual) educational community, whereas proprietary systems tend to be very unspecific about this question. Collective responsibility can count as a true motivational factor ("Together we are strong") and for instance is demonstrated by the success of the Wikipedia project [2001]. To conclude, we indicated that the transformation from recognition of potential to actualising it is not automatic, and moreover, that the User Riddle had to be expected under an educational and design view. A user's motivation to invest time (and money) into educational software cannot be derived from its potential alone. In [Kohlhase & Kohlhase, 2004a] we pointed out that added-value services are one way to specifically enhance the value of an author's investment and we already hinted at invasive technology. Elaborating on latter under a design perspective, we will now take a closer look at more of users' realities as "a designer has to have a solid understanding of the complexity involved in being rational" [Löwgren & Stoltermann, 2004, p. 50].

3. Invasive Technology

The User Riddle is tackled from several scientific directions at the moment. For instance, general requirements for ICT acceptance are described in the "Nine Rules for Good Technology" by Stephen Downes [2004, pp. 11]. Others include user profiling, building specific niche solutions, adapting to different learning styles, community-building, automating services like indexing, and "easy-to-use" user interfaces (see projects like e.g. [Wikipedia, 2001], [Mumie, 2001], ActiveMath in [Melis et al., 2001], Ariadne Foundation in [Duval, 2004]). Our approach consists in basing the design of new ICT functionalities on the users' realities. ICT shapes not only the work space but also leisure space, the distinction dissolves more and more. Media designer Lev Manovich speaks of a "gradual computerization of culture" [2001, p. 6] and points out [sic, p. 215]: "increasingly the same metaphors and interfaces are used at work and at home, for business and entertainment".

In particular, it implies that a user is confronted with ICT all along. Since a user's time capacity is commonly very limited, every user should only have to master one software tool for each basic ICT skill like editing, calculating, or organizing. The design scheme of "same look and feel" for different functionalities (e.g. realized in Office Suites), also called consistency principle, supports this requirement as well.

Moreover, this implies that the non-technical properties of software like being pleasurable have to come into view (see e.g. [Vyas & van der Veer, 2005]). According to interaction designers Jonas Löwgren and Erik Stolterman, "function, structure, and form have to be juggled in an interlocking balance. Design is never only one or the other of these aspects, but all of them at the same time." [2004, p. 54], therefore composition is a designer's real work. In particular, the whole set-up has to make the user comfortable as we want the user not only to use ICT in the long run but live with it (see [sic, p. 163]). Hence, we argue that the user has her own software preferences and a designer of a new ICT module has to take this seriously - independently of the ICT designer's judgment on the preferred software.

A direct consequence is the motto of the architecture of "invasive technology": Let the user choose her preferences for software covering basic ICT skills and enhance this software by new components. The new technology so-to-speak "invades" already existent (and used) software. Concretely, we call an editing facility an invasive editor, if it is build into an existing application and performs neglected functionalities like content markup. Such an add-on is nurtured by the existing editor in the sense that it adopts its feel, look, and location. As an example I will present the semantic editor CPOINT for content in MS PowerPoint.

3.1. CPOINT as Invasive Editor

In particular, we concerned ourselves with capturing knowledge in MS PowerPoint (PPT) lectures, i.e. semantically enhancing the content of slides, so that it can be shared and reused as a learning object over the Web. Being specifically well aware of the User Riddle with respect to content authoring e.g. for the Semantic Web, we implemented the content authoring tool CPOINT (Content in PowerPoint), that allows to annotate (and manage) objects on a slide from within PPT, as an invasive editor within the scope of the Course Capsules Project at Carnegie

Mellon University, USA [Kohlhase et al., 2002].⁴⁹ The task at hand can be illustrated by the following situation: A teacher is accustomed to using a software tool like PPT to transform her implicit knowledge into explicit knowledge for a presentation to a class; if she wishes to enhance this knowledge (by semantically annotating it), she will perceive leaving this editor as painful. Her motivation for wanting to provide open content (based on the underlying potential of digital media for education) can be destroyed if she not only had to leave her accustomed tool for doing so but also if she had to learn yet another tool for the basic task of editing text. This clearly presents a hurdle for her willingness to generate open content and can be alleviated by providing semantic editing facilities within the chosen proprietary editor, i.e. an invasive editor.

To be more specific, CPOINT (written in Visual Basic for Applications as a PPT add-in) makes its functionality available through a toolbar in the PPT menu (see Figure 1) where it is at a user's disposal whenever the PPT editor is running. Moreover, the added functionalities are reached within the "same look and feel"-frame as other PPT tools.



Figure 5: The CPOINT Menu Bar

The integration with the open source world is realized by CPOINT's conversion function. Here, the enhanced PPT presentation can be converted into other formats, specifically the semantic XML standard OMDOC [Kohlhase, 2004]. These generated OMDOC documents can be fed to eLearning systems like ActiveMath - enabling students to make additional use of a teacher's presentation. Moreover, users may aggregate new presentations by picking content from available OMDOC documents - enabling e.g. teachers to reuse and share their knowledge. Once a user starts employing the new facilities, she is supported by added-value services, tipping the trade-off scale towards the worthiness of the user's investments.

The invasive technology idea is further realized within CPOINT's math user interface (MUI) [Kohlhase, 2004b]. We extended the PPT math editor Texpoint [Necula, 2003] by a semantic math editing facility. Texpoint itself enables a user to use a Latex-like input style for math in PPT, it is therefore itself invasive technology. It is broadly used within the scientific community. But it is based on glyph tables (to optimize presentation) and not on mathematical symbols (to optimize mathematical knowledge management), so we developed a Texpoint-style and -based MUI that fully integrates mathematical symbols into PPT presentations based on the underlying semantic objects rather than simply generating appropriate ink marks on the screen. In Figure 2 we see the Latex-style input where the "macro" names like "ourPlus" are the underlying symbol names which were assigned a PPT-representation format by the user.

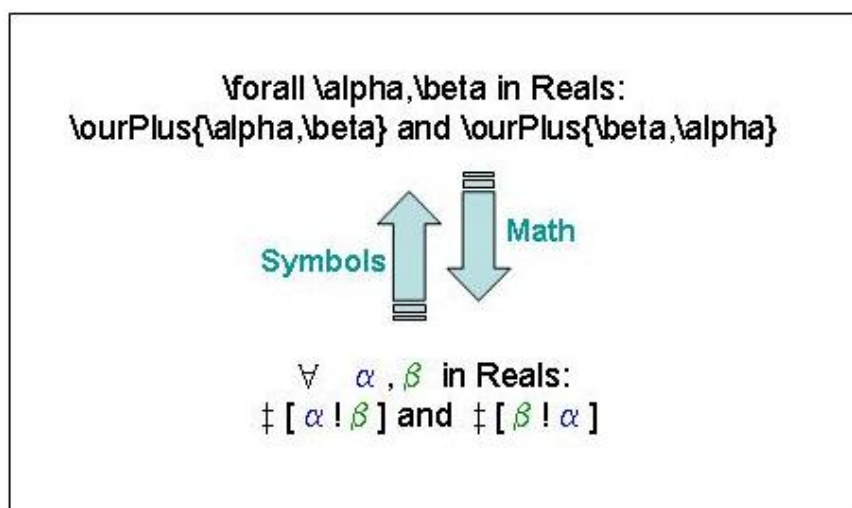


Figure 6 Effects of CPOINT's Semantic Math User Interface

⁴⁹ It is distributed under the Gnu Lesser General Public License, the newest version can be downloaded from <http://kwarc.eecs.iu-bremen.de/software/>

4. Conclusion

We exemplified the contrast between scientific expectations with respect to educational ICT and actual users' acceptance by a description of the Semantic Web's history. This User Riddle was analyzed by the respective expectations from scientists on the one hand and users on the other and found them to be quite different. The scientific perspective is based on the futuristic potential of ICT whereas a user's is based on her past experiences and current situation. Therefore the integration of both points of view is necessary. Our integrative approach is based on "Invasive Technology", i.e. software that makes use of already accepted technology by invading it and being available from within, offering new functionalities whose potential is seen by scientists as well as users. We presented CPOINT as invasive editor in proprietary MS PowerPoint in order to produce open content – overcoming proprietary software as hurdle for user acceptance of open source software for education.

References

- Anderson, T., & Whitelock, D. (2004). The Educational Semantic Web. Viewed on 2005/01/02 at www-jime.open.ac.uk/2004.
- Berners-Lee, Tim (1998). What the Semantic Web can represent. <http://www.w3.org/DesignIssues/RDFnot.html>.
- Berners-Lee, Tim, & Fischetti, Mark (1999). Weaving the Web: The original design and ultimate destiny of the World Wide Web, by its inventor. Harper San Francisco.
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web - Computers navigating tomorrow's Web will understand more of what's going on making it more likely that you'll get what you really want. *Scientific American*, 284.
- Bray, T. (2003). The RDF.net Challenge. Online on 2005/10/05 at <http://www.tbray.org/ongoing/When/200x/2003/05/21/RDFNet>
- Downes, Stephen (2004). The Learning Marketplace: Meaning, Metadata and Content Syndication in the Learning Object Economy. Online Book. Viewed on 2005-10-06 at <http://www.downes.ca/files/book3.htm>.
- Duval, Eric (2004). We are on the Road to In L. Cantoni & C. McLoughlin (Eds.), Proceedings of the ED-MEDIA 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 3-5. CD-ROM.
- ERA (2005). Building the ERA for Knowledge and Growth. online. Seen at 04-07-2005 at <http://www.rp6.de/inhalte/rp7>.
- Frauenfelder, M. (2005). Sir Tim Berners-Lee. Online on 2005/10/04 on <http://www.technologyreview.com/articles/04/10/frauenfelder1004.asp>.
- Heid, Helmut (2004). Kann man zur Verantwortlichkeit erziehen? Über Bedingungen der Möglichkeit verantwortlichen Handelns. In Johanna Hopfner & Michael Winkler (Eds.). Die aufgegebene Aufklärung: Experimente pädagogischer Vernunft, Beiträge zur pädagogischen Grundlagenforschung, pp. 145-154. Juventa Verlag Weinheim und München.
- Kohlhase, Andrea & Kohlhase, Michael (2004a). CPOINT: Dissolving the Author's Dilemma. In Andrea Asperti, Grzegorz Bancerek, & Andrej Trybulec (Eds.). *Mathematical Knowledge Management, MKM'04, LNAI*, 3119. Springer Verlag.
- Kohlhase, Andrea (2004b). CPoint's Mathematical User Interface. Workshop Math User Interfaces at MKM04. Viewed on 2005/10/07 at http://www.activemath.org/~paul/MathUI/proceedings/CPoint/CPointMathUI%_MathUI04.pdf
- .

Kohlhase, Michael, Sutner, Klaus, Jansen, Peter, Kohlhase, Andrea, Lee, Peter, Scott, Dana, & Szudzik, Matthew (2002). Acquisition of math content in an academic setting. MathML02, Second International Conference on MathML and Technologies for Math on the Web. Chicago.

Kohlhase, Michael (2004). OMDoc: an Open Markup Format for Mathematical Documents (Version 1.2). Manuscript, <http://www.mathweb.org/omdoc/omdoc1.2.ps>.

Löwgren, Jonas, & Stolterman, Erik (2004). Thoughtful Interaction Design: A Design Perspective on Information Technology. The MIT Press.

Manovich, Lev (2001). The Language of New Media. The MIT Press.

Melis, E., Andres, E., Franke, A., Goguadse, G., Libbrecht, P., Pollet, M., & Ullrich, C. (2001). ActiveMath System Description. In Johanna D. Moore, Carol Luckhard Redfield, and W. Lewis Johnson (Eds.), Artificial Intelligence in Education, 68, p. 580 – 582. IOS Press.

Mumie (2001). Platform for Learning and Teaching Mathematics. Online at <http://www.mumie.net/>.

Necula, George (2003). Texpoint 2.0.3. Online Documentation. Viewed on 2005/10/07 at <http://raw.cs.berkeley.edu/texpoint/TexPoint.html>.

Schluss, Henning J. (2002). Bildungstheoretische Kriterien der Lehrplananalyse: Ein Diskussionsangebot an die Politikdidaktik. In Reinhold Hedtke (Ed). Onlinejournal für Sozialwissenschaften und ihre Didaktik, 1, 2002. Viewed on 2005/06/02 at http://www.sowi-onlinejournal.de/2002-1/lehrplananalyse_schluss.htm.

Vyas, Dhaval, & van der Veer, Gerrit C. (2005). Experience as 'Meaning': Creating, Communicating and Maintaining in Real-Spaces. Online Interact 2005 workshop proceedings. Viewed on 2005/09/26 at www.interact2005.org.

Wikipedia (2001). The biggest multilingual open access encyclopedia on the internet. Online at <http://www.wikipedia.org/>

From learning design model to automated synthesis of a web-based educational application: A CASE tool

Symeon Retalis ¹, Andreas Papasalouros ², Vaggelis Skouteris ¹, & Kostantinos Siassiakos ¹

¹ University of Piraeus

Department of Technology Education and Digital Systems

80 Karaoli & Dimitriou 185 34 Piraeus, Greece

retal@unipi.gr, vskouteris@unipi.gr, siassiakos_k@ypan.gr

² National Technical University of Athens

Software Engineering Laboratory

9 Heroon Polytechneioy 157 80 Zografou, Greece

andpapas@softlab.ntua.gr

1. Motivation

Design and implementation of web-based educational applications are complex, if not overwhelming activities. This is due to the fact that they involve people from diverse backgrounds such as software developers, web application experts, content developers, domain experts, instructional designers, user modelling experts and pedagogues, to name just a few. All these people are trying to describe and build the presentational, behavioural, pedagogical and architectural aspects of the courseware.

The phases of a web-based educational application development process (i.e. instructional system development process) are identical in all development projects. The eight phases, as illustrated in Table 1, are (or should be) performed by various people with specialised skills and competencies.

Phases	Activities	Outcome
Analysis of the context	<ul style="list-style-type: none"> – Assess learning problems/needs – Assess user/target population – Define constraints and restrictions imposed to the solution finding process 	<ul style="list-style-type: none"> – Objectives, syllabus, assessment – Context of use – Development environment
Design of the web-based educational application	<ul style="list-style-type: none"> – Development of architectural design – Courseware components specification – Interface design – Navigation design – Prototyping 	<ul style="list-style-type: none"> – IMS Learning Design – Prototype Interface design
Creation-Development of Components	<ul style="list-style-type: none"> – Preparation of multimedia content – Preparation of textual content – Preparation of (inter)active components 	<ul style="list-style-type: none"> – Resources & Metadata – Services
Formative evaluation	<ul style="list-style-type: none"> – Pilot testing – Analysis of data gathering 	<ul style="list-style-type: none"> – Feedback report
Integration/Packaging	<ul style="list-style-type: none"> – Integration of the various elements into a whole 	<ul style="list-style-type: none"> – Web-based educational application (SCORM)
Summative evaluation	<ul style="list-style-type: none"> – Pilot testing with real learners, tutors 	<ul style="list-style-type: none"> – Evaluation study
Deliver & Maintenance	<ul style="list-style-type: none"> – Maintenance for correction – Maintenance for perfection 	<ul style="list-style-type: none"> – Versions of web-based educational application

Table 1. The phases of web-based educational application's development process

The most well known process model is the prototyping or evolutionary model (Alessi & Trollip, 2001). We, as developers, believe in the added value of this evolutionary model. It suggests that the application is divided into components, or increments, which cover part of the functionality of the overall product and part of the overall requirements. The developer designs in detail, builds, formatively evaluates and produces the application, increment by increment. The learner's perception of the overall educational application end product is a moving target. As increments are being built and delivered, learners' and teachers' opinions regarding the value of the remaining increments may change. Features of the applications felt to be essential to the overall product may be deleted or new increments may be added to the list of increments to be incorporated into the overall product during some future development phase. The advantage of this model is the improvement of manageability and visibility (IMS, 2001) of the development process by allowing users to give feedback quite often during the process. Furthermore, this model allows the addition of courseware components that have not been specified and included in the courseware design from the beginning. In fact, the evolutionary model does not distinguish between developing a product and enhancing (maintaining) it; enhancement is merely an additional increment that is built. Nowadays, almost everyone who is involved in web-based educational application development projects talks about standards. Most probably, this means that the development process (or technology) is maturing and there is a need for standards compliance. In the context of learning technology, standards are still evolving while some of them are quite stable such as IEEE LOM (IEEE, 2005) and IMS QTI (IMS, 2005). Trying to identify how standards are correlated to the various development phases, a mapping, which can be found in Figure 1, can be created.

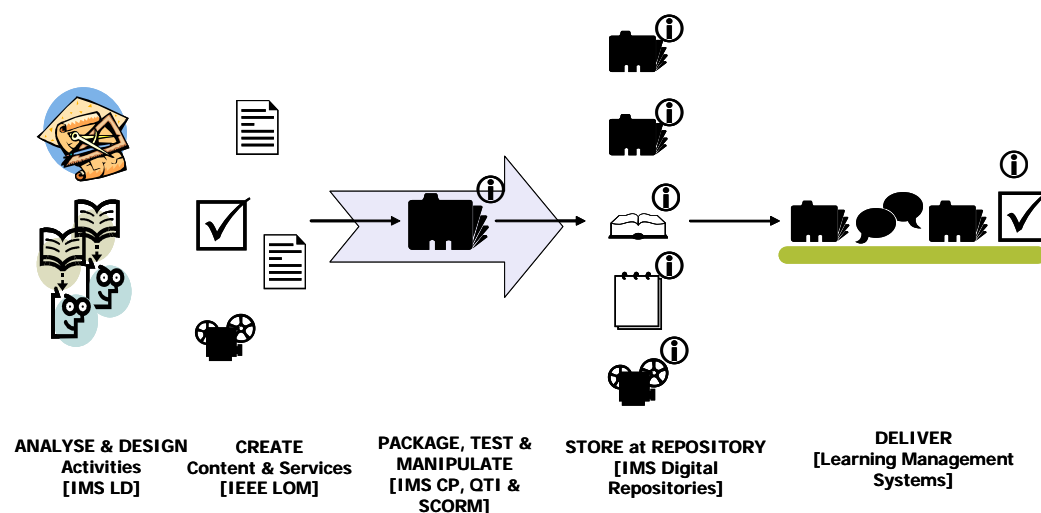


Figure 1. Phases and learning technology standards in web-based educational application's development process

There are several tools that try to support developers in performing the activities of each phase guaranteeing, at the same time, that the outcome of these activities will be compliant to the standards. Unfortunately, these tools are disjoint, meaning that they do not embrace the whole development process but just isolated phases. Thus, there is an emerging need to create tools that can support the developers as much as possible in their effort to design and build web-based educational applications compliant to the standards.

Some efforts have started by consortia working in projects such as Reload (Milligan, Beauvoir and Sharples, 2005), Unfold (<https://www.unfold-project.net:8082/UNFOLD/>), eXe (<http://exelearning.org/>) projects. Details about the LD-tools are given in (Koper and Tattersall, 2005). Isolated attempts by groups have given outcomes such as the Collage (<http://ulises.tel.uva.es/collage/>) and LAMS (<http://www.lamsinternational.com/>) tools. A few attempts have been made by researchers that work in the field of web engineering such as (Aroyo et al., 2002), WebML, WCML, UWE, etc. as described in (Retalis & Papasalouros, 2005).

In this paper, we present our approach for bridging the gap between the conceptual description of the solution to an instructional problem (i.e. learning design) and its implementation as a web-based educational application. We have developed a design method for web-based educational applications, which will be briefly described in section 2, and a tool

for supporting the developers (see section 3). The design method and tool has been applied in several projects by teachers for building educational applications in various subject domains. There are various things that can be added into the functionality of the tool which will be discussed in the final section of the paper.

2. The design and authoring process

CADMOS-D, as a design method, provides both a process and a product model (13, 19) for educational web applications development. The former pertains to the detailed description of the various design steps, their temporal relationships and sequencing, and the description of their outcomes, while the latter refers to the specification of the outcomes of each step which capture the design decisions for the application under development, the relationships and dependencies between these outcomes and the mechanisms that allow these outcomes to drive the development of the actual application. Furthermore, the product model can form the basis for the description of existing applications, provide the blueprints that diffuse knowledge and common understanding for particular applications, either completed or under development, much in the way that the blueprints of a building can both drive its development and depict its form, structure and function. The Unified Modelling Language (15) has been used as a visual notation and modelling language for the design model of CADMOS-D. More specifically, the UML-profile described in (Papasalouros & Retalis, 2005) has been developed for the definition of the modeling language of CADMOS-D. However, the product model conforms to the learning technology specifications.

More specifically, the process model of CADMOS-D has the following characteristics. First, a conceptual description of the educational application is provided as a solution to the instructional problem under consideration. This conceptual description has been derived from the IMS-learning design specification as well from the field of instructional design. It mainly consists of a map of interrelated activities associated to learning objects and learning services used by actors (learners, instructors, etc). The learning activities are associated with specific reusable learning objects which are both individuals or composites as well as learning services offered by software systems.

Moreover, two other separate views of the educational application must be mentioned for the above description to be complete. One is the navigation view, which refers to the allocation of learning activities into hypertext nodes, i.e. web pages, as well as the management of hyperlinks between these nodes. These links also follow the well known categorisation of links in hypertext literature: structural links, that derive from the structure of the material, and associative links, that implement domain specific relationships between the interlinked pieces of information. The other view is that of the presentation or user interface design of the educational material, that is, the specification of the presentation aspects of the educational material and the comprised learning objects. A graphical representation of the aforementioned approach is shown in figure 1. Details about the CADMOS-D method can be found in (Retalis & Papasalouros, 2005).

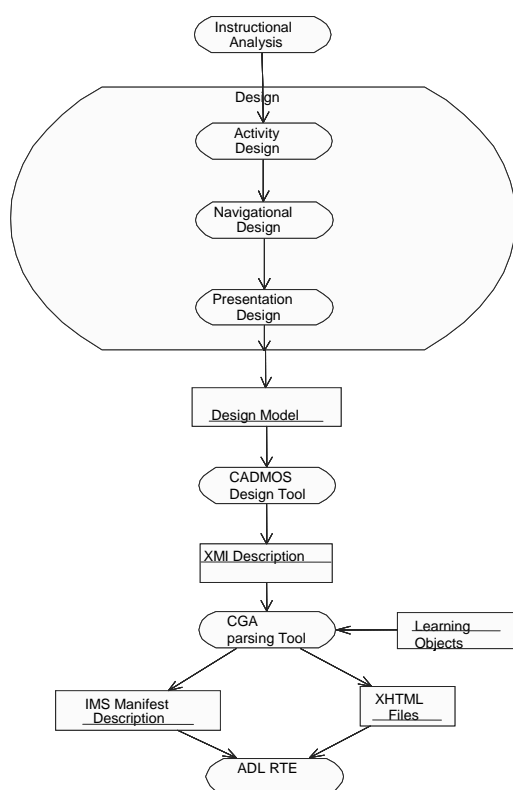


Figure 1. The steps for designing and automatic synthesising a web-based educational application

The outcomes of the CADMOS-D method are of two kinds: Prototypes of educational applications, that is, complete but unfinished in their details applications but also constructs of learning objects of higher level of granularity than their constituting parts that have resulted from the proposed process. Constructs of learning objects are described in two ways: a) the IMS-learning design specification of the educational application specification and b) the educational application, which conforms to the IMS-Content packaging.

3. A Case Tool – CADMOS Processor

Although CADMOS-D can be supported by any UML-CASE tool, a specialised tool for the design and automatic packaging of the actual educational resources has been developed for supporting the whole approach. The tool is called CADMOS Processor. This tool was specially designed such that teachers and instructional designers with limited computing skills and basic knowledge of the learning technology standards, could use it. Before presenting the tool, let's see how CADMOS-D can be supported by a UML-CASE tool. Imagine that the designer has experience in UML and CASE tools like Rational Rose. The UML-design models will hold the details of the three aforementioned views of the educational application. As an example, consider an educational application about Fire Safety (an online tutorial). According to the designer, the learner should study some introductory material about the learning process and the material. Then, he/she should solve a pre-test for estimating his/her knowledge level and misconceptions. In the sequence, the learner will access to online material about various topics of the subject domain, e.g. how to use a fire extinguisher, based on the results of the pre-test. Using the CADMOS-D primitives and the UML-notation, the three views/models of the application could look like the designs of Figures 2a, 2b, 2c.

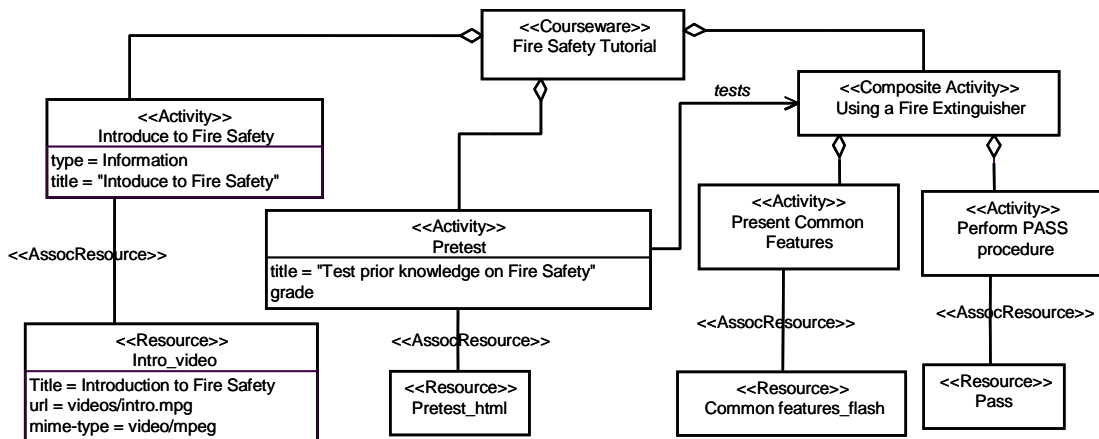


Figure 2a. The UML-based learning activity design model of the fire safety tutorial

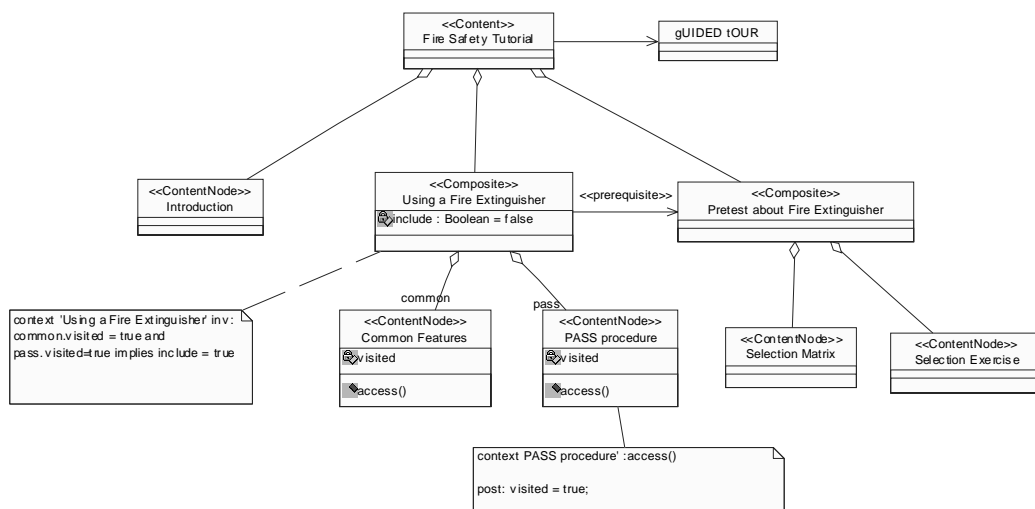


Figure 2b. The UML-based navigational design model of the fire safety tutorial

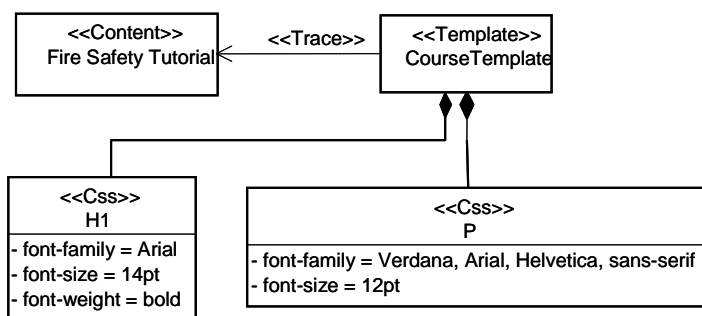


Figure 2c. The UML-based presentational design model of the fire safety tutorial

The model presented previously with the Rational Rose tool is exported in XMI-format, the OMG standard XML metadata interchange format (OMG, 2003) for describing UML- models. With the use of a specially developed tool, called CGA (Courseware Generation Application), the UML model is transformed from XMI to a structured hypermedia educational application conforming to the IMS Content Packaging standard. More specifically, the CGA-tool accepts as entry the XMI-description with the relevant learning objects (HTML- pages, pictures, files of

sound and video, active objects as Applets, ActiveX, Flash, etc.) and produces as output the real educational application.

Although, the above diagrams are comprised of symbolic shapes and lines, it is not easy for a teacher with basic computing skills to use a UML Case tool. He/she is more accustomed to using concept map building tools, for their instructional design. This is the reason why we built the CADMOS Processor. This tool hides the UML notation formalism details and offers a user friendly lay-out to the teacher in order to create the learning design and the navigational diagram. At the back-end of the tool, these diagrams accord to the specificities of the CADMOS-D primitives. Figures 3a, 3b, 3c show screen shots of the tool and the design models that can be created.

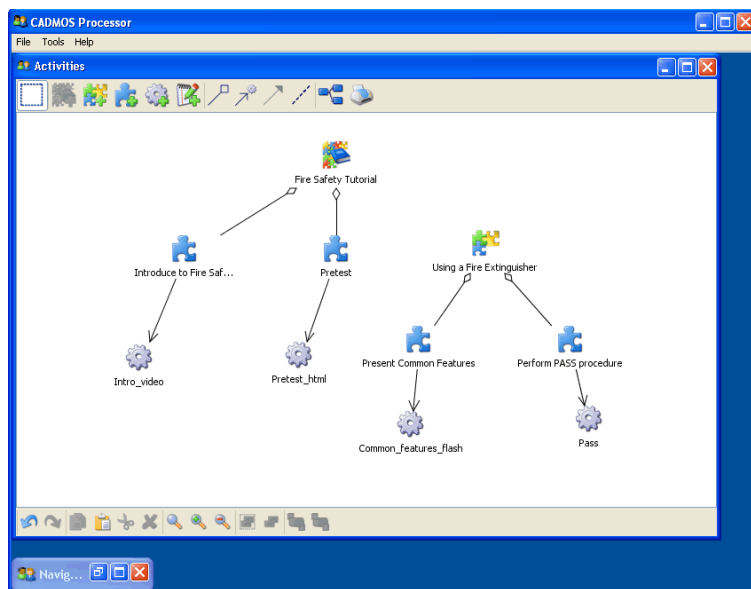


Figure 3a. The learning activity design model of the fire safety tutorial using CADMOS Processor

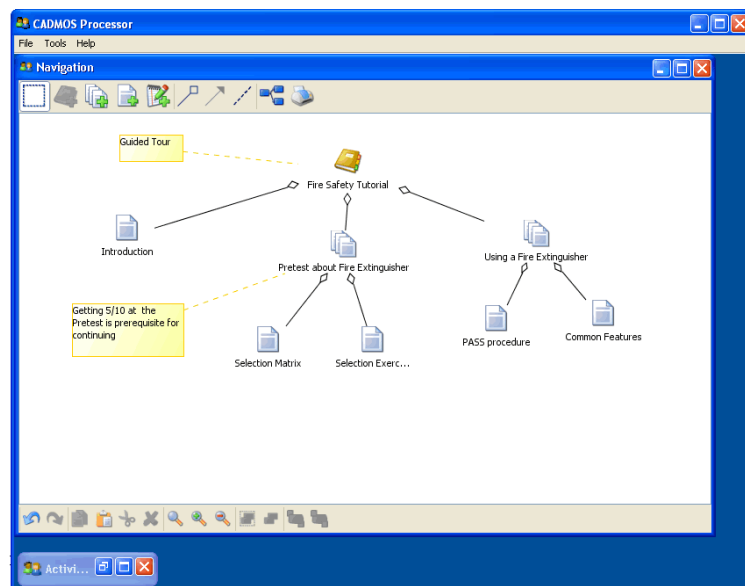


Figure 3b. The Navigational design model of the fire safety tutorial using CADMOS Processor

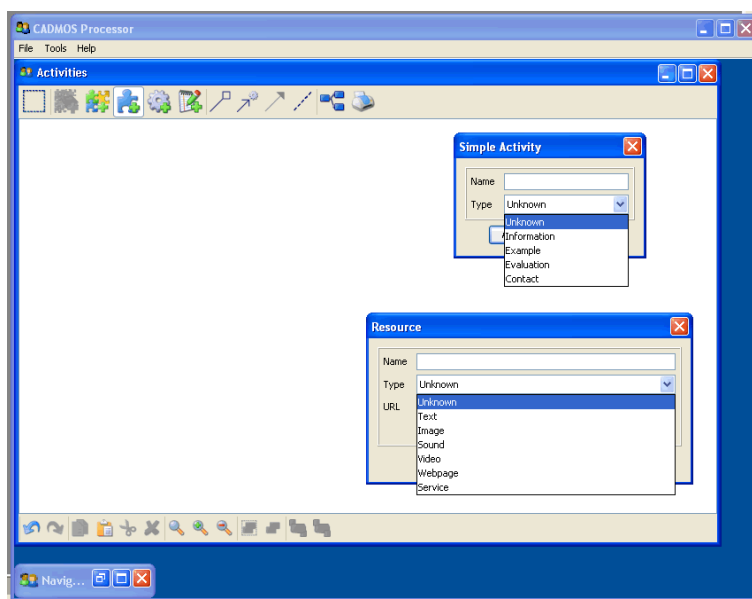


Figure 3c. Screen shots from the CADMOS Processor that show the pop-down menus for adding a simple activity and annotating its type as well as inserting a resource of a specific type

Cadmios Processor was developed as a Java application, which means that it can run in a variety of operating systems. The graph creation and management is based on the JGraph toolkit (<http://www.jgraph.com>) which offers rich graph editing and layout capabilities. CADMOS Processor can give as output the model of the application in the IMS-LD compliant format as well as the application as an IMS-CP compliant package. It can be easily figured out that CADMOS Processor cannot, at the moment, support the designer in the interface design. Default style sheets can be used for the packaging. Or experienced designers can manually link some pages to specific Cascading Style Sheet files.

4. Conclusions

In this paper the CADMOS-D design method has been outlined. We tried to bridge the gap between conceptual design and prototyping of a web-based educational application. A tool that supports this method, called CADMOS Processor, has been analysed. This tool still needs enhancement. It is among the top priorities to add two extra features: i) the sketching of interface design models and their linkage to the pages that appear at the navigational model, and ii) the creation of appropriate diagrams which specify rules for personalised sequencing of learning resources based on the definition of dynamic navigation behaviour that will be transformed into IMS Simple Sequencing schema (IMS, 2003b) as supplement to the content packaging description, which is incorporated in the Learning Design description of the educational applications. The latter feature will look like UML activity design diagrams already prescribed by the CADMOS-D method. We believe that methods and teacher-friendly tools, that advocate i) the separation of concerns in the design of web-based educational applications, thus dividing the design of the application in three stages: conceptual, navigational and presentational, and ii) the traceability of decisions in each stage, will augment the involvement of more teachers in the actual production of applications that will comply to the learning technology standards.

Acknowledgement

This work was partially supported by the “TELL: Towards effective network supported collaborative learning” project (<http://cosy.ted.unipi.gr/tell/>) which is partly sponsored by the European Commission under e-learning program (ref num: 2003-4721 TELL).

References

- Alessi, S.M. & Trollip, S.R. (2001). *Multimedia for Learning: Methods and Development* (3rd ed.). MA: Allyn and Bacon.
- IMS (2005). IMS Question and Test Interoperability Overview - Version 2.0 Final Specification, retrieved from <http://www.imsglobal.org/question/>, October 13, 2004.
- IMS (2003a). IMS Simple Sequencing Information and Behaviour Model Version 1.0 - Final Specification, retrieved from <http://www.imsglobal.org/simplesequencing/>, October 13, 2004.
- IMS (2003b). IMS Learning Design Information Model Version 1.0 - Final Specification, retrieved from <http://www.imsglobal.org/learningdesign/>, October 13, 2004.
- IMS (2001). IMS Content Packaging Information Model Version 1.0 - Final Specification, retrieved from <http://www.imsglobal.org/contentpackaging/>, October 13, 2004.
- Koper, R., and Tattersall, C. (eds.) (2005). *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, C., Springer: Berlin/ Heidelberg/ New York.
- Milligan, C.D., Beauvoir, P. and Sharples, P. (2005). The Reload Learning Design Tools. In: *Journal of Interactive Media in Education (Advances in Learning Design. Special Issue)*, 2005/08. (ISSN:1365-893X).
- Papasalouros, A. & Retalis, S. (2005). Defining a UML Profile for Web-based Educational Applications. In: Daniel Schwabe, Gustavo Rossi, Luis Olsina, Vicente Pelechano (eds): *International Workshop on Web Oriented Software Technologies, Proceedings of the CAiSE2005 Workshop on Web Oriented Software Technologies, Porto, Portugal, June 13, 2005. CEUR Workshop Proceedings, ISSN 1613-0073, online <http://CEUR-WS.org/Vol-153/>.*
- Retalis, S., & Papasalouros, A. (2005). Designing and Generating Educational Adaptive Hypermedia Applications. In: *Educational Technology & Society*, 8 (3), p. 26-35.

Open Source Software in Teaching Physics: A Case Study on Vector Algebra and Visual Representations

Erdat Cataloglu, Ph.D.
Physics Education, Faculty of Education
Abant Izzet Baysal University
erdat@ibu.edu.tr

Mustafa Baser, Ph.D.
Physics Education, Faculty of Education
Abant Izzet Baysal University
mbaser@ibu.edu.tr

Abstract

This study aims to report the effort on teaching vector algebra using open source software (OSS). Recent studies showed that students have difficulties in learning basic physics concepts. Constructivist learning theories suggest the use of visual and hands-on activities in learning. We will report on the software used for this purpose. The effect of OSS on students understanding of vector algebra was determined by a non-equivalent control group design. A total number of 113 freshman students from two classes of introductory level physics courses were involved. The experimental group's students learning processes were supplemented by instruction utilizing OSS while the control group was taught in traditional manner. A significant difference in students' performance was found that could be attributed to the software used. Consequently, visualization of vector and related concepts by OSS simulations helped students to understand them well and contributed to shorten the time needed to learn these concepts.

Introduction

Teaching and learning is a complex process which is being studied intensively. Important studies regarding human learning constantly effect learning theories especially those related to school learning. Learning models based on behaviourism are now being scrutinized. Many problems have been identified with traditional teaching approaches where the teacher is the authority and the students are passive learners. This type of structure favours the world view of "one type correct" answers, even to complex problems (e.g. environment & pollution), and has an elitist approach towards students. While some students are able to perform and solve complex problems in physics they fail to apply basic knowledge in novel situations (Driver et al., 1994).

Recent theories focusing on the nature of learning promote the constructivist theory. This theory arose out of Piaget's works in developmental psychology. Briefly, constructivism regards the learning processes as a continuous construction and reconstruction of concepts. Throughout these procedures the students are actively involved in their own learning processes. Students are put into a situation where an engaging environment is being created by stimulating, challenging, and provoking the interest of them. Teaching strategies should be structured in such a way that the students are active participants and the teachers act more like a guide rather than the all knower. Constructivist teaching and learning theories emphasize that teaching should be build around open ended problems where students are allowed to explore different paths to reach desired conclusions. Another important factor to consider is that learning takes place in social environments. Therefore, peer interaction is viewed as an essential component in cognitive development (i.e., learning) (Feltovich et al., 1996).

Accordingly, contemporary learning theories emphasize the need to provide the learners with a variety of learning opportunities. These environments should also include the possibility of peer interaction and collaboration opportunities.

Rational

Reaching a meaningful understanding of Vectors is essential to learn the concepts presented in physics, algebra and geometry. In physics, vectors are defined as physical quantities that have both magnitude and associated direction to it. Displacement, velocity, acceleration, force, momentum, and impulse are all examples of vectorial quantities. Unlike scalar quantities such as temperature, mass, time, energy, power, and work the mathematical manipulation of vectors is somewhat more complicated. For a typical introductory mechanics course the topics to be studied include areas such as kinematics, dynamics and Newton's Law of motion, work and energy, impulse and momentum, and rotational motion. In order to reach a sound understanding of the concepts presented in these topics a basic understanding of vector algebra is also needed.

Online tutorials and aids can help the students to improve their understanding of vector mathematics by providing immediate feedback in a structured environment. Rothney, Roselli and Howard (Rothney et al., 2003) developed software called "Courseware Authoring and Packaging Environment" (CAPE) that supplies a diagnostic correction mechanism that identifies common student errors and provides specific feedback based on the type of mistake encountered for a biomechanics course. This study aims at investigating the effect of providing a curriculum that makes use of a series of additional tools to help the students understand the concepts of vector algebra for introductory mechanics. Visualization was a key factor while deciding on these tools. Interactivity was another important factor. Finally we also opt for tools that would be available to the students both in class as well as out of class. This way we also gave the students the opportunity to work outside of class with their peers. Although we did not obligate collaboration.

For this study we utilized non-commercial software for obvious reasons. We wanted to build an additional learning tool for our students next to what they have been already using. As it is in many introductory physics classes, students use books and to some extent calculators depending on the level of physics being taught. Apart from these tools, the students have nothing else, especially outside of class hours, to help them in the learning process of vector mathematics. Students' misconceptions in vectors can hinder their ability to advance in successive topics in mechanics because the concepts presented in kinematics, dynamics, momentum & impulse, and rotational motion all depend on a sound understanding of vector algebra. The software used was Octave, GoOctave.cgi, Java applets, GNU-plot, putty, Ssh & bash and an Apache web server all running on a Linux OS. A somewhat extended explanation will follow on Octave, GNU-plot and the Java applets whereas a discussion on the latter software (i.e., Apache, ssh, bash etc.) will be omitted since we believe they are all well known. More information about Octave follows as it relates to vector algebra.

Octave

Octave is a powerful mathematics tool. In some references it is also called a high-level language. It is a tool with a command line interface as its main source of interaction and is especially designed to manipulate matrices. The tool can be used to accomplish numerical computations as well. MatLab, for example, could be considered as a commercial equivalent of Octave. In fact, the description in its LSM-entry reads "GNU Matlab--A numerical matrix mathematics program."

Octave was designed as a pedagogical instrument to help teach better issues around chemical reactors and problems related to that subject. The first version was released in 1994 and since then has undergone numerous revisions. The authors "wanted to create something that would enable students to solve realistic problems, and that they could use for many things other than chemical reactor design problems". Octave is now included in Debian GNU/Linux and SuSE Linux distributions, rpm-binaries for RedHat and Fedora redistributions. Hence Octave seems to have reached a major success "Today, thousands of people worldwide are using Octave in teaching, research, and commercial applications". (<http://www.octave.org/history.html>)

Information on how to use Octave can be easily found at their website. They provide an extensive manual on how to use the tool. A WIKI-website provides additional help topics ranging from how-to's of simple compiling and installation issues to complex scripting and advanced batch programming. Finally, as mentioned before, Octave is similar to Matlab, so references to tutorials regarding Matlab are also to a great extent valid for Octave. Especially those that use the command line interface as the main input and manipulation interface.

GoOctave.cgi

The goOctave.cgi script is written by Mai Zhou (<http://www.ms.uky.edu/~statweb/testOctave.html>). The script gives the students an interface with Octave over the web through a simple html-form. By means of such an interface the students also have direct access to Octave out of class. Students can run examples and experiment with the examples from anywhere they have web access. Another major advantage of employing this script is to have the tool ready for teaching and provide students who do not own a computer with Octave.

The GoOctave.cgi script provides the students with a form where they can type the equations. The result is shown on the same webpage. Moreover, GoOctave.cgi can render also graphs (in png-format) if desired. The GoOctave.cgi script had some limitations which were easily overcome. The authors made some coding modifications to meet our needs. For example, the script could not run in multiple-user mode. To count for this, the code was modified such that it could render graphs while simultaneous users were working with the webpage. Additionally all descriptions were translated into Turkish, the native language of the students. The webpage can be accessed through the following URL: <http://per.ibu.edu.tr/cgi-bin/goOctave.cgi>. In figure 1 we display the interface.

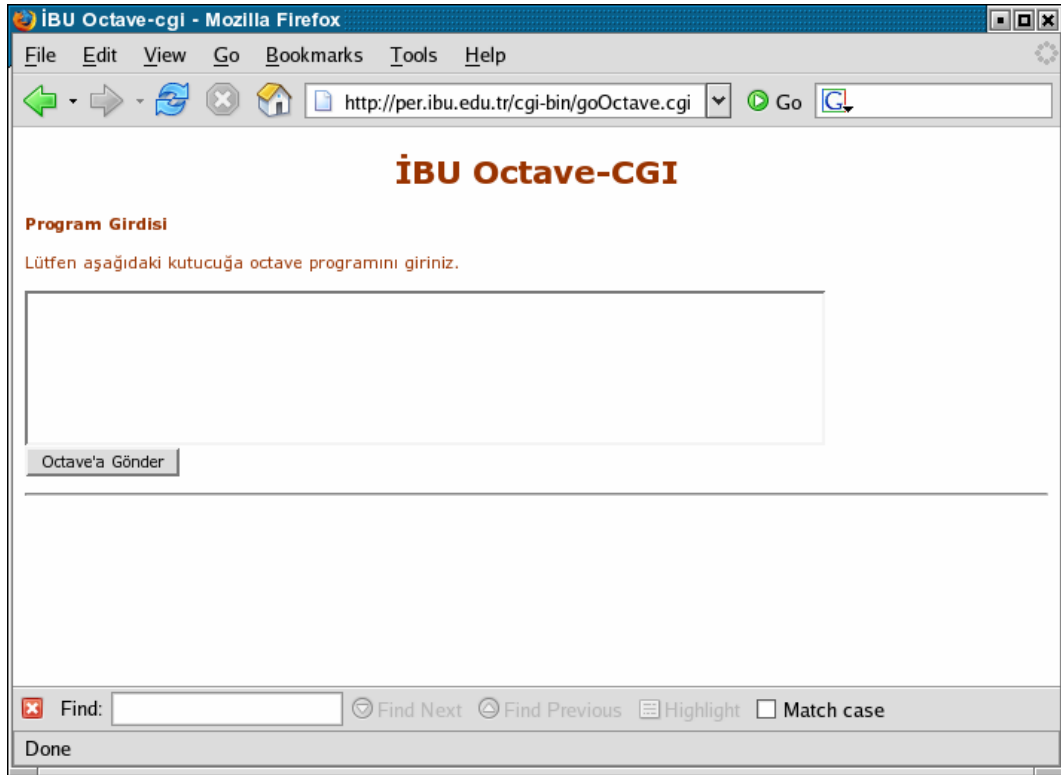


Figure 1. Modified Web Interface of goOctave.cgi

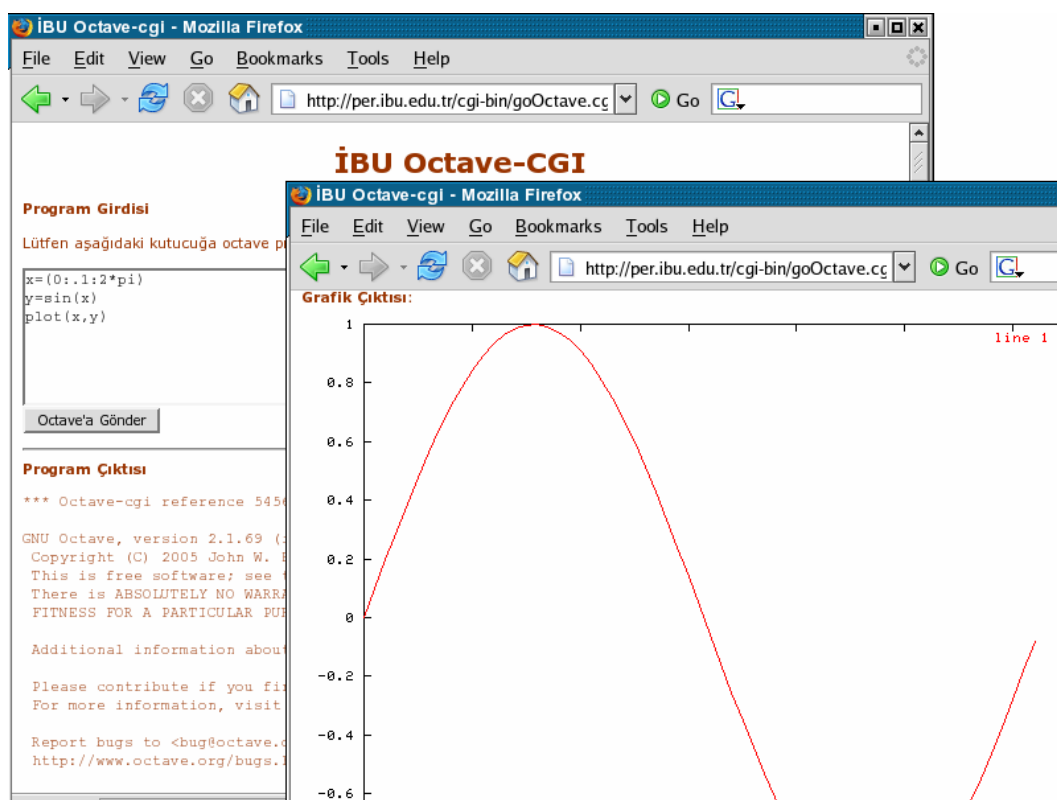


Figure 2. Web Interface: Sample Output

Example on Basic Vector Algebra and Octave

Since vectors can be expressed as column matrix Octave is an ideal candidate for basic vector algebra.

Vectors are simple to enter into Octave. To input a one dimensional vector one needs to type in the following text

$$a = [0,3;0,0]$$

a is a vector parallel to the x-axis that is 3 units in length

$$b = [0,0;4,0]$$

b is a vector parallel to the y-axis that is 4 units in length Now adding these two vectors is a simple task. One only needs to type in the following expression:

$$c = a + b$$

In order to compute the magnitude of the new vector c, one has to type the following expression:

$$d = \text{mag}(c)$$

here 'mag' stands for magnitude which is a slightly modified function of trace; a build-in function of Octave. For the above example Octave returns the magnitude of the vector addition as:

$$d = 5$$

Java Applets

Java applets provide powerful visualization aids especially in physics education. Since the Java technology has emerged, many applets have been created for physics instruction. These applets range from simple cases to complex modelling of physical phenomena. For example a very complex modelling example of a real time 3D physics simulator applet can be found at <http://www.ambromley.co.uk/fizz.html>.

Due to the numerous benefits Java Applets can bring into a physics classroom and its relative easiness to employ have lead to many applets available for physics instruction. Physlets resource pages (<http://webphysics.davidson.edu/Applets/Applets.html>) is one such

example. Many interactive examples ranging from introductory level physics to quantum mechanics can be found at this site. In figure 3 such a vector applet is shown.

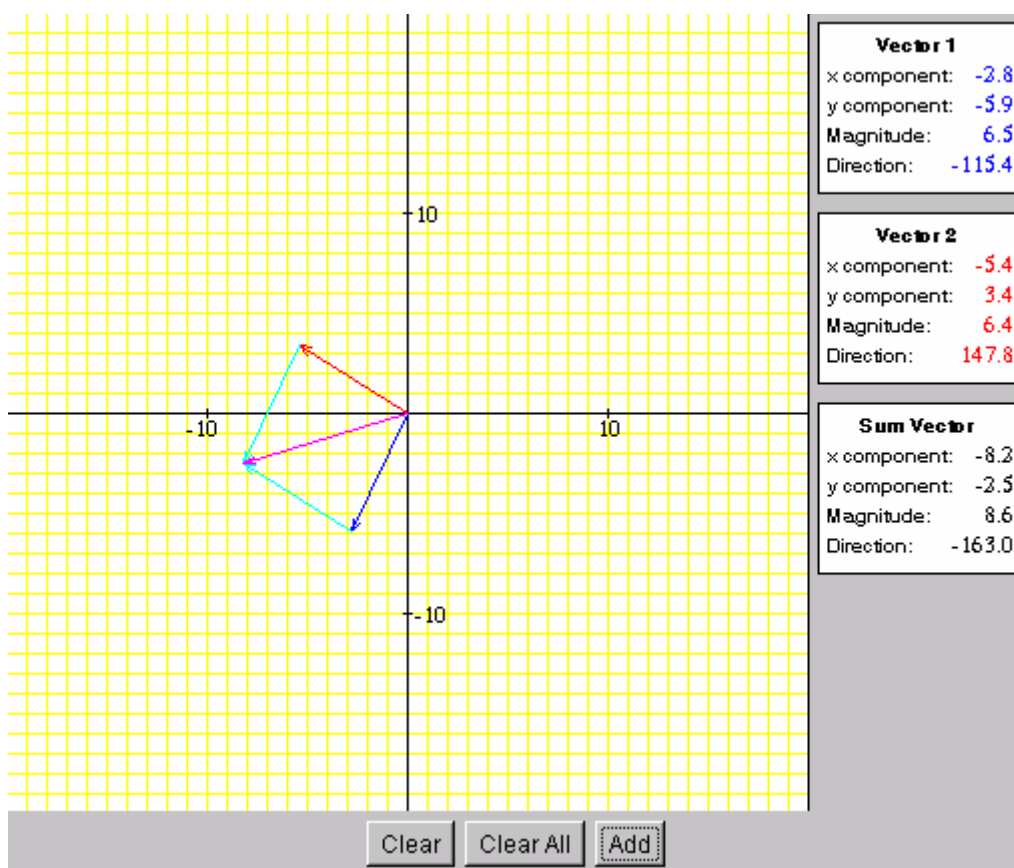


Figure 3. Vector Applet

Vpython

Vpython and python is other open source software used in this case study. Python⁵⁰ is an object oriented programming language that allows quick implementation of an idea in an easy and clear syntax. As such, Python enables people to focus on programming rather than bothering about syntax of language. Visual is a 3D graphics engine that works as loadable module which offers real-time 2D- and 3D-output with very little coding needed and reasonable performance. Python + Visual is called Vpython.

Stajano (Stajano, 2000) explored how Python can be used in education. Elkner (Elkner, 2000) discussed the use of Python in a high school computer science program. More related references and material related to Python and education can be found at <http://www.python.org>.

The University of Illinois' Physics Education Research Group (PERG) uses Vpython interactive programs to promote students' understanding of physics concepts. VPython is used as teaching material in the course PHYSICS 271 at the Department of Physics of the Purdue University. Rob Salgado uses Vpython for teaching physics. Rob's programs can be downloaded at <http://physics.syr.edu/~salgado/software/vpython/> free of charge. While teaching the concepts of motion in one and two dimensions we used the Kinematics programs to reinforce the vectors of the physical properties of position, velocity, and acceleration.

⁵⁰ <http://www.python.org>

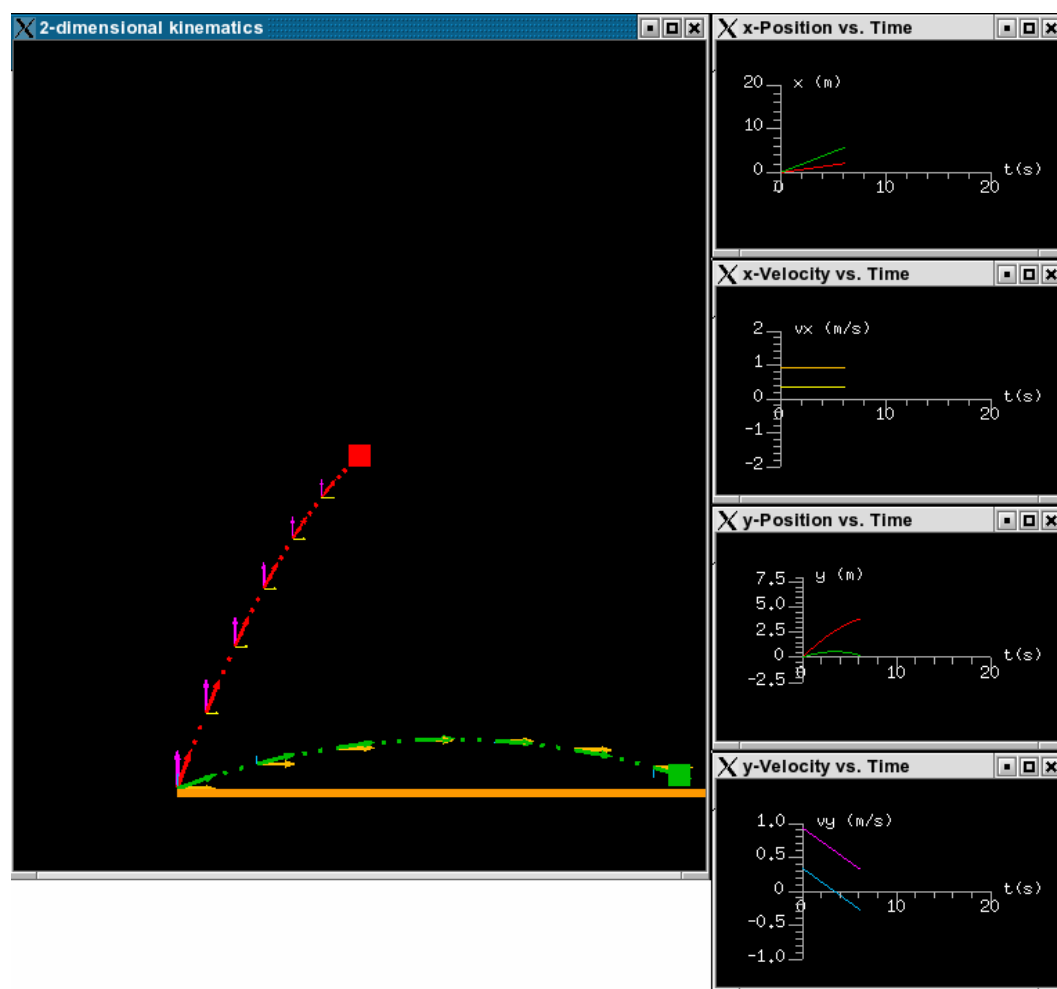


Figure 4. Vpython – Projectile Motion

In this program two projectiles with different initial angles and velocities are projected. For both objects the students are able to analyse the projectile path in detail. The projectiles' vectorial motion is being represented by colored arrows, which enhances the visual conception of the concepts presented.

Method

The study took place during the Summer School of 2005 at Abant İzzet Baysal University. A total number of 113 students who participated in the study were divided into two groups, an experimental group (55 members) and a control group (58 members). General Physics 1 is a one semester long mandatory freshmen-level introductory mechanics course with three 50-minute lectures per week and no laboratory work. Most of the students took the course for the first time (68%) (the rest had to redo the course). The number of female students was slightly more (62) than the number of male students (51). A five open-ended question pretest was administered at the beginning of the educational unit of vectors. The same test was administered as a posttest at the end of the kinematics unit. The students were asked to do basic vector calculation which they were expected to encounter on a regular basis for the forthcoming topics. In the first question the students had to add two vectors. With the second question the students should find the components of a given vector. The third question was about computing the magnitude of a vector. Finally, in the fourth and fifth question the students had to perform a dot product and cross product respectively.

The experimental group received a handout detailing on how to use the web interface, java applet, and how to use the command line interface to login into the server where Octave was installed. Additionally, during office hours assistance was provided for students who needed it.

Results

The data for this study were the pre-and posttest scores of the 113 freshmen students. The data were analyzed by using R which is a GPL-licensed language and environment for statistical computing and graphics (<http://www.r-project.org/>). The web interface for R, provided by Montana State University (<http://bayes.math.montana.edu/cgi-bin/Rweb/Rweb.cgi>) was used to analyze the data.

To test if there was a significant difference between the experimental and the control group prior to instruction, a two sample t-tests was run on the pretests scores. The result from RWeb is as follows:

Welch Two Sample t-test

```
data: EXPPRE and CNTPRE
t = -0.4685, df = 110.445, p-value = 0.6403
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-5.393516  3.330820
sample estimates:
mean of x mean of y
16.72727  17.75862
```

As can be seen from the RWeb result, there was no statistically significant difference between control and experimental group prior to instruction.

To test the effect of two types of instruction on students understanding of vector algebra, the two samples t-test was run on the pretests. The result from RWeb is as follows:

Welch Two Sample t-test

```
data: EXPPOST and CNTPOST
t = -0.9058, df = 65.097, p-value = 0.3684
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-154.60450  58.11965
sample estimates:
mean of x mean of y
32.36364  80.60606
```

This result showed that there was statistically significant difference between post test scores of students in experimental group and control group. Thus, it can be argued that, utilizing Open Source Software in teaching vector algebra is more effective than a traditional teaching method for this case. Further experiments should be done to verify this result and for making further generalisations.

In addition, to the vector algebra test, an open ended questionnaire was administered to the experimental group after the instruction. The questionnaire questions and the common answer given by students are given in Table 1

<i>Question</i>	<i>Common Answers</i>
Did you like or dislike the software used in this course. If your answer is 'No', please specify why.	58% of the students liked the software. Some of the students thought that it was somewhat difficult to use
Do you think the software used in this course is easy to use? Please explain your ideas.	47.3% of the students stated that the use of the software was not easy.

<i>Question</i>	<i>Common Answers</i>
Would you recommend this instruction for the next semester's students in a freshmen physics course? If your answer is 'No', please explain why?	40.0% of the students recommended it for the new students. 34% of the students stated that, software used in this course should be modified to make it more user-friendly.
Do you think the software used in this course increased your understanding of vector algebra? If your answer is 'No', please explain why?	56.4% of the students answered YES to this question. 16.4% of the students didn't perceive any value using this tool; moreover they felt that it did not make any substantial contribution to their understanding of vector algebra.

Table 2: Questionnaire and Answers

Although, some students had negative attitudes toward the utilisation of the software, still an important number of the students' answers were relatively positive (see Table 1).

Conclusion and final remarks

Ann Thompson points out the importance of FOSS: "open source software will provide new and exciting possibilities for educators. Obviously, the free or low-cost availability of open source software has great appeal for educators at all levels, and the Linux system is already gaining popularity in schools and districts around the country [USA]. Equally important to the cost issue, however, is the opportunity provided by open source software for education to adapt software to the needs of their students. In the same way a teacher might adapt a lesson plan to his or her needs, open source software may provide the opportunity to adapt a software program" (Thompson, 2002).

In this study we attempted to provide our students with the ability to use advanced FOSS in order to help them to get a better understanding of the concept of vectors. Needless to say, a sound understanding of vector algebra in introductory mechanics is crucial and will help the students further towards a better understanding of the concepts presented in topics such as kinematics and dynamics. It is our belief as science educators, that we are responsible for providing our students with as many learning tools as possible. Octave and GoOctave.cgi are examples of such learning tools, they are readily available, and are relatively easy to setup. Although, not part of the goal of this study, we also experienced that by using Octave through its command line interface lets the students develop understanding of another aspect of computers, namely that computers are actually computing devices that do recursive tasks, beyond the students accustomed point and click use provided by so many GUI applications. The use of a web-based application is also important, because it provided access to the computer software outside the school laboratory facilities. This way the educators could provide the students with educational tools, in our case with GoOctave.cgi and Java vector applet, which gave them the ability to study and work outside of class hours. Therefore, FOSS helps to foster the use of asynchronous teaching methods. Clearly, one can envision that the deployment of FOSS towards specific needs will facilitate broadening the classical learning environment which will incorporate asynchronous teaching methods beyond traditional Learning Management Systems which are more generic in nature.

Visualisation through Java Applet technology was also beneficial. This technology provided instant visual aids like how the resultant vector with respect to the vector components changed, for example. The students liked the software, especially the Java applet. Overall, we feel that, although some negative feedback was received by the students, we added some extra, valuable learning tools for the students.

References

- Driver, R., Squires, A., Rushworth, P., & Wood-Robinson, V. (1994). *Making sense of secondary science: Research into children's ideas*. New York: Routledge.
- Elkner, J. (2000). 'Using Python in a High School Computer Science Program'. Appears in: *Proceedings of 8th International Python Conference, 24-27 January 2000, Washington, D.C.* (available at <http://www.python.org/workshops/2000-01/proceedings/papers/stajano/stajano.html>).
- Feltovich, P., Spiro, R., Coulson, R., Feltovich, J. (1996). 'Collaboration with and among minds: mastering complexity, individually and in groups'. In: *CSCL: Theory and Practice of an Emerging Paradigm*. Mahwah, NJ: Lawrence Erlbaum, Associates.
- R: A Language and Environment for Statistical Computing, R Development Core Team, R Foundation for Statistical Computing, Vienna, Austria, 2005.
- Rothney, M.P., Roselli, R.J., & Howard L.P. (2003). 'Creation of an online vector addition tutorial: Exploring the advantages of providing diagnostic, multilevel feedback in basic skills remediation'. In: *ASEE Annual Conference, (CD-ROM DEStech Publications) Session 2793: 7 pages, 2003.*
- Rothney, M.P., Roselli, R.J., and Howard L. (2003). 'Creation of an online vector addition tutorial: exploring the advantages of providing diagnostic, multilevel feedback in basic skills remediation'. A paper presented at American Society for Engineering Education.
- Stajano, F. (2000). *Python in Education: Raising a Generation of Native Speakers*. Appears in: *Proceedings of 8th International Python Conference, 24-27 January 2000, Washington, D.C.* (available at <http://www.python.org/workshops/2000-01/proceedings/papers/stajano/stajano.html>).
- Thompson, A. (2002) 'The Open Source Software Movement: Implications for Teacher Educators'. In: *Journal of Computing in Teacher Education*, p.110.

Using IMAP to Build a Virtual Learning Environment

Michael Klebl

Faculty of Philosophy and Education – Ergonomics and Industrial Pedagogy

Catholic University of Eichstätt-Ingolstadt, Germany

michael.klebl@ku-eichstaett.de

Abstract

Undoubtedly, ease-of-use is a critical factor in the success of information systems in education and training. Hence, high effort is needed to build sophisticated web-based graphical user interfaces for virtual learning environments. This paper presents the initial idea to leave the beaten track of HTTP/HTML in building the client-server-interaction between a learner and learning management system in favour of using IMAP, an open and increasingly common protocol for accessing e mail. The way in which common concepts of VLEs map to the use of a modern e mail-client as a well-known user interface for learners and teaching staff is discussed as well as resulting advantages and disadvantages. A use case will start from the idea of users having an e mail account to log on, where they find all necessary information organised in mailboxes, folders and e mails. A mock-up user interface that could be implemented by the OS-community is presented. This draft is based on IMS Learning Design, since this interoperability standard covers a comprehensive framework for learning scenarios.

Introduction

A virtual learning environment (VLE) is an information system used in education and training which provides personalized access to educational content, communication services and interactive tools. From the viewpoint of a learner, a VLE should allow effortless comprehension of course outlines as well as the opportunity to navigate a course and offer a consistent and intuitive user interface when moving from one course to another. For this paper, a VLE is understood as the user interface for learners and teaching staff used in learning scenarios. A VLE thus provides access to resources and services operated by a learning management system (LMS). Most VLEs are web-based and thus to be accessed through any web browser. In addition to the World Wide Web, e mail is the most used application of the internet. If you would ask people (at least somewhat experienced users) what application software they start first and shutdown last when working with a personal computer, they are likely to name their e mail client. E mail clients are used to organise personal information such as contacts and notes along with mails and documents. Some e mail clients offer additional features for personal information management, such as calendars and task lists, or are integrated in software for collaborative work, i.e. groupware. So if a web browser can be used to access a VLE, why not think of accessing information for education and training with an e mail client. This paper presents the idea of having an e mail client as a user interface for learners and teaching staff. The following screenshot (Figure 7) presents this idea from the perspective of a music student enrolled in some music courses (for the exemplary unit of learning see Tattersall & Burgos, 2005):

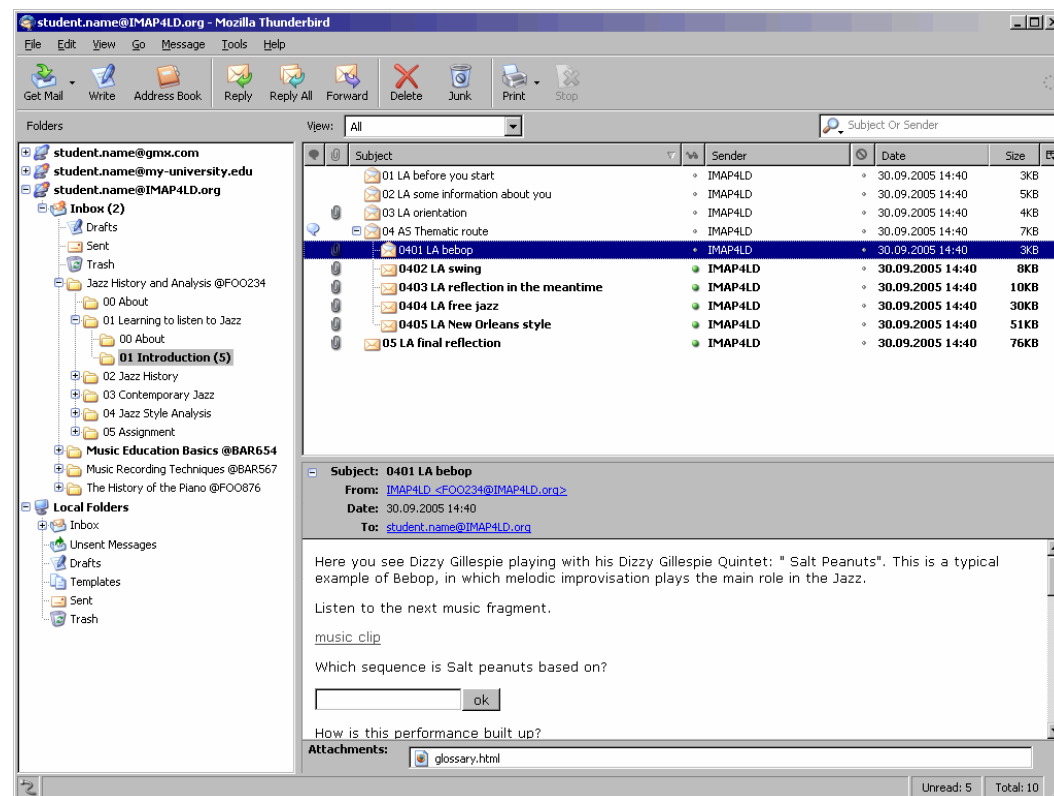


Figure 7. Screenshot for access to a VLE using an e mail client

This music student has set up an extra e-mail account for access to a VLE, where he or she finds message folders for each course in which he or she is enrolled below the inbox folder. Having an e-mail client as a user interface for a VLE presents certain advantages:

- Some users use their e-mail client as their daily workspace. Most users of personal computers use their e-mail client daily. Whoever has an e-mail address will use an e-mail client regularly.
- An e-mail client is basically a cooperative software application. Like any communication tool, it connects people to other people and thus creates a social setting.
- An e-mail client is both a rich and native software application. It offers a wide range of features and functions for managing information and often raises less security issues compared to web browsers.
- A wide range of e-mail clients exist for different devices (like personal computers, PDAs or smartphones). Accessing e-mails is platform-independent based on a few, widely acknowledged standards.

One of these standards is IMAP (Internet Message Access Protocol, RFC 3501, see Crispin, 2003). IMAP is an internet protocol used for accessing e-mail on a remote server from a local client. Since IMAP is in the application layer of the internet protocol stack, it can be used instead of HTTP in order to establish a connection between a user of a VLE and the server hosting this VLE. Together with the Internet Message Format (RFC 2822, see Resnick, 2001) for basic message format plus encoding as well as MIME (Multipurpose Internet Mail Extensions, RFC 2045 through RFC 2049, see Freed & Borenstein, 2001a, 2001b) for enhanced message content, IMAP offers a set of required functionalities for client-to-host communication that can be used to provide access to a VLE.

Client-to-host communication for access to e-mail with IMAP is based on authentication and personalisation. With the support of online, offline and disconnected mode, IMAP supports synchronisation for information objects on a server especially for mobile clients (see Mullet & Mullet, 2000). Since IMAP is designed to access a remote mailbox with similar functionalities as if e-mails and mailboxes were stored locally, it builds on basic principles which allow advanced organisation of information (see Mullet & Mullet, 2000):

- For internet mail, messages are the basic unit of information. Messages considered as nodes are structured in a specific way. They consist of a header and body. Using MIME, messages as information nodes can be structured further, allowing

alternative parts, attachments, internal references and even tree structures. With IMAP, messages have flags that hold status information similar to file attributes.

- IMAP supports management of remote message folders, i.e. mailboxes on a server, which hold a collection of messages. Since mailboxes can be nested, tree structures can be built similar to file systems. In an IMAP session, mailboxes have a particular function: In the selected state a single mailbox is chosen. Information about the content of this mailbox is retrieved by the client. Within a single mailbox, messages are sorted by a message sequence number. Furthermore, messages can be arranged in tree structures through the threading of messages (using header information 'In-reply-to:').

Given these functions incorporated in concepts of internet mail and implemented in e-mail clients, this paper presents the initial idea to build the client-server-interaction between learner and learning management system on IMAP and related specifications for internet mail. The way in which common concepts of VLEs map to the use of a modern e-mail client as a well-known user interface for learners and teaching staff during the running of a course is analysed in Section 0. For an outlook on implementation in Section 0, basic interactions of a user with a VLE provide a few use cases, where the look and feel of an e-mail client as a graphical user interface is to be examined. Only a mock-up for an implementation is presented. Issues regarding advantages and disadvantages of a possible implementation are discussed. This initial idea of a new interface for a learning management system could be enhanced further through discussion and could be implemented by the OS-community. Not within the scope of this paper is the specification of administration and authoring features for courses and educational content to be attained by the use of an e-mail client.

This draft for the use of IMAP and related specifications for communication between client and VLE refers to IMS Learning Design, since this interoperability standard covers a comprehensive framework for learning scenarios (see Koper, Olivier, & Anderson, 2003 part 2.1). Some basic concepts of IMS Learning Design have to be considered, although the basic idea of this paper may apply to VLEs and learning scenarios not compliant to IMS Learning Design. For further reading on IMS Learning Design, a comprehensive article can be recommended (see Koper & Olivier, 2004). Basic ideas and concepts of IMS Learning Design relevant for this paper may be summed up as follows:

- Fundamental to the specification IMS Learning Design is the concept of 'units of learning'. Common terms for this concept are 'course', 'module' or 'lesson'. A unit of learning consists of a description of learning objectives, prerequisites, learning activities, teaching activities and services along with an outline of the teaching-learning process as well as resources for learning used in this process (or at least references to these resources). Complexity and granularity of a unit of learning cannot be determined in general. However, a unit of learning is a self-contained period in a teaching-learning process, limited in time and dedicated to a certain issue of a subject that is studied (see Koper, 2001). For the purpose of this paper, we assume that for instance a university course spanning a whole term covers a number of units of learning (i.e. 'plays' in IMS Learning Design). They may be sequenced as kinds of chapters in the course. Within the weekly organisation of university courses, each of them may span one or several weeks, depending on learning objectives, subject matter and arrangement of the teaching-learning process.
- Since IMS Learning Design is dedicated to the paradigmatic shift from description of content to description of process for a unit of learning, its information model is based on activities. In general, an activity is something to be done by someone in order to achieve a purpose. In a learning context, activities are directed towards learning objectives. Learners perform learning activities while teaching staff perform supporting activities. Both use resources for learning, e.g. educational media or tools, and both interact and communicate directly or via special devices, i.e. both use web-based services. Learning activities (as well as supporting activities) may be marked as 'completed'.
- An outline for the teaching-learning process is assembled through aggregation of activities (not as aggregation of content). IMS Learning Design indicates special levels of aggregation. A learner realizes a sequence of phases ('acts') within a unit of learning. These phases are clearly separated sections. Within a section, there are activities assigned to a role. While a learner does this or that, the teacher does another activity. For example: While learners do an exercise, teachers provide help. Performing an activity can be structured by container elements named 'activity structure'. Thus, on the lower level of aggregation, a learner realizes a sequence of

activities (there might also be a choice of activities). Structured activities assigned to roles form a learning flow, i.e. an outline for the teaching-learning process.

- In order to perform these activities, the learner finds resources and services in the designed 'learning environment'. In IMS Learning Design, learning environments are an important element to order and structure learning objects and services used in the teaching-learning process. Environments are container elements. Learning objects and services cannot be assigned to activities directly, so learning environments are linking elements between activities and resources for learning. But learning environments are not only linking elements. Instructional designers may use them to group learning objects and services into packages, in order to reuse these packages and to present them to the learner at a given step in the teaching-learning process.

Other concepts of IMS Learning Design could be applied to the notion of this paper, but cannot be discussed in detail here. For example, the aggregation of learning activities within a unit of learning is completed by a description of learning objectives and prerequisites. Furthermore, Level B of IMS-LD adds three concepts which allow the creation of more elaborate learning flows within a unit of learning: properties, conditions and monitor. Properties (like services) have to be instantiated at runtime and record outcomes of activities during operating a unit of learning. Given logical and arithmetic expressions in combination with if-then-statements in the condition element, modification in the learning flow can be planned and executed during runtime on the basis of properties. By a special service called 'monitor', learners or teaching staff may be allowed to access information stored in properties, in order to enhance interaction between learner and teaching staff or within a group of learners.

The aggregation of single activities within container elements such as activity structures, phases (in LD-terms 'acts') and methods (in LD-terms 'plays') is represented by an activity tree. Tree-based organisation of informational elements is very common to graphical user interfaces ranging from file systems to web-based information systems. As shown above, tree-structure organisations are also inherent to IMAP and related specifications for internet mail.

Mapping Concepts of VLEs to IMAP

As stated above, a VLE provides access to resources and services operated by a learning management system (LMS) and is used by learners and teaching staff. In order to examine the notion of using an e-mail client as a well-known user interface for learners and teaching staff, common concepts of VLEs are described in the first part of this section. A mapping to the concepts of IMAP and related specifications is presented in the second part, based on concepts of IMS Learning Design.

1.1. Common concepts of Virtual Learning Environments

Virtual Learning Environments usually are defined as information systems that integrate services for computer-mediated communication and interactive, computer-based tools with digital delivery of resources for teaching and learning. Virtual Learning Environments are thus seen as an integrative information system used by both learners and teaching staff. For the purpose of this paper the use of VLEs in learning scenarios is of main concern. Requirements for administration courses and authoring educational content are not taken into account. Requirements focussing on access for learners and teaching staff to resources, tools and services provided by a learning management system (LMS) can be summarised as follows (see Britain & Liber, 1999):

- *Personalized Access*: A VLE has to offer personalized access to educational content, communication services and interactive tools. Only learners registered at an educational institution such as a university have access to information and services. Learners usually enrol in courses or learning communities and thus have personalized access to resources for learning and services for communication. Hence, authentication of a user against a VLE is a basic requirement for a VLE, including when providing a personalized user interface.
- *Access to Learning Objects*: A VLE has to offer access to resources for learning. Resources for learning can be manifold: texts, figures, exercises, lectures, simulations, experiments, problem statements and others are considered as learning objects. However, a VLE has to offer easy access to learning objects within a given learning context such as a course running at a moment in time. Within learning scenarios, users of a VLE (learners as well as teaching staff) require a

suitable and comprehensible structure of information provided by the VLE based on authentication and personalization.

- *Access to Services:* Learning scenarios that use a VLE often use computer-mediated communication within a group of learners or between learners and teaching staff alongside with distribution and perception of learning content. Hence, a VLE has to provide services for interaction within a group of learners and for different supporting roles of teaching staff, like tutoring, coaching and informing. As for learning objects, access to services has to be easy and within the context of a course or a learning community. Thus access to services is based on authentication and personalization as well.
- *Personal Desktop:* A common concept for easy and personalized access within a VLE is a personal workspace, where an authenticated user finds educational content, communication services and interactive tools. Often users can change this personal desktop, e.g. by adding objects such as bookmarks or icons for access to content, services and tools to it.
- *Course structure:* Within learning scenarios (independently of whether a VLE is used or not), resources for learning as well as learners' tasks (in the sense of learning activities) are organised within a course structure. A course structure is an outline for the teaching-learning process and may include conditions for transition from one phase in the learning flow to another, sometimes depending on assessment. Hence a VLE has to represent a course structure to the learner not only as a way of navigating within a learning scenario, but also in order to provide orientation about the process of learning.
- *Tracking Information:* In addition to the outline of a course structure representing a teaching-learning process, a VLE has to record information about activities performed by learners (as well as by teaching staff). This tracking of information by the VLE provides feedback to the learner, e.g. what step in the process of teaching and learning is current, what has already be done and what is to be done next.

From the viewpoint of a learner, far more requirements could be described, such as a *personalised learners' portfolio or timetable and calendar features*. Furthermore, administration and authoring features would open a wide area of requirements that are to be considered. However, if a VLE offers highly contextualised, personalized access within a course or a learning community and within a given step in the teaching-learning process, it definitely will suitably meet requirements of learners and teaching staff during the running of a course.

1.2. Mapping to Concepts of IMAP

From the viewpoint of learners and teaching staff in the context of a learning scenario, personalised access to resources for learning that is contextualised within a course or a learning community is a major requirement for a VLE. While in the use of HTTP for client-to-host communication personalisation has to be achieved through sessions, either using cookies or parameters in URLs, authentication is inherent to any session using IMAP. With IMAP, data can be organised in messages as basic units of information. Messages within single mailboxes are sorted and marked by flags. Tree structures for information can be represented by nested message folders (i.e. mailboxes), by threading of messages (using header information 'In-reply-to:') within a message folder, and by the internal structuring of messages using multipart messages with MIME. Given a logon to an IMAP account, message folders and messages as basic elements, a mapping to concept of a VLE for IMS Learning Design can be made as follows:

- An **IMAP account** fulfils the notion of a *logon to a VLE*. A learner is given an IMAP account for access to a VLE in an educational institution and just the once has to create a new mail account within his or her e-mail client for this access.
- **Message folders** in this IMAP account represent *different courses or learning communities*, in which a learner is enrolled. At a second level, message folders represent *single units of learning* within a course (i.e. 'play' in IMS Learning Design). At a third level, message folders represent *phases* in the teaching-learning process (i.e. 'act' in IMS Learning Design). Since in IMS Learning Design activities or activity-structures can only be assigned to roles within an act using multiple role-parts and thus the transition from one act to another applies to all roles, acts serve as a synchronization element (Koper et al., 2003 part 4.4). Hence the selection of a message folder representing an act with the IMAP command SELECT fits the notion of performing structured activities during a particular phase in the process of teaching and learning.
- A single **message** corresponds to an *activity* to be performed by a learner (learning activity) or by teaching staff (supporting activity). An activity description is included

as body of this message, hence displayed in the e-mail client when this activity is selected. Tree structures of activities ordered by activity structure elements are built using the threading of messages in a message folder. Subordinate activities have to be marked with an 'In-reply-to:' header field referring to the identifier of the activity structure to which they belong. Therefore, messages also represent *activity structure* elements. The sorting of messages within a single mailbox by the message sequence of IMAP corresponds to the succession of activities and activity structures in the process of teaching and learning.

- **Multipart messages using MIME** can be used in different ways to provide *access to elements in the learning environment*, i.e. to learning objects and communication services used in order to perform learning activities. Single documents that are used as learning objects can be supplied as attachments, using the MIME-Header 'Content-Disposition: attachment'. For learning objects, interactive tools and communication services that are to be used interactively in a web browser, only URLs are supplied. These URLs can be given in the simple text part of the message, as most e-mail clients will convert them into hyperlinks. A collection of URLs could also be represented in an HTML part of the message, allowing the designing of learning environments with icons for elements to be used for performing activities. Both ways of providing links to elements in the learning environment would use the MIME-Header 'Content-Disposition: inline', in order to be displayed directly in the e-mail client as part of the e-mail body.

For particular functions of VLEs the mapping of concept between IMAP (including related specifications) and a VLE cited above could be extended even further. During a practicable implementation, far more ideas must be developed. Only two more applications shall be mentioned here:

- The message folder **INBOX** used in any IMAP account should hold current and personal information; hence the inbox of a learner's account corresponds to a *personal desktop*. Thus, all pending activities from current units of learning are listed within the inbox. Unquestionably, other current news items are placed in the inbox as well.
- Some IMAP servers offer **shared folders** that can be used equivalent to newsgroups, thus as *asynchronous communication services*. Other than the message folders used for a course outline described above, shared folders with read/write access serve as newsgroups for interaction within the group of learners or with teaching staff. They allow users logged on to the VLE to exchange information depending on their enrolment in courses or learning communities.

For a quick overview, the following **Error! Reference source not found.** lists all suggested mapping between IMAP and a VLE:

Concepts of IMAP	Concepts of VLEs
IMAP Account	Logon to a VLE
Message Folders	Different Courses or Learning Communities Single Units Of Learning (i.e. 'play' in IMS-LD) Phases (i.e. 'act' in IMS-LD)
Message	Activity Structure Activity
Multipart Messages using MIME	Access to Elements in the Learning Environment, i.e. Learning Objects as Attachment linked through a URL Services linked through a URL
INBOX	Personal Desktop
Shared Folders	Asynchronous Communication Services

Table 3: Mapping of common concepts for a VLE to concepts of imap and related specifications

Implementation

In this section, an outlook is given for a possible implementation of the idea presented in this paper. Firstly, a few basic interactions of a user with a VLE provide use cases, where the look and feel of an e-mail client as a graphical user interface can be examined. Secondly, a possible implementation of an IMAP Interface for a learning management system that could meet the requirement of these use cases is roughly outlined. A further elaboration of advantages and disadvantages ends this section.

3.1 Use-Cases and GUI

In order to illustrate the idea proposed in this paper, the mapping of common concepts for interaction of users with a VLE to the concepts of IMAP and related specifications is applied to a few use cases. These use cases are defined by three basic interactions of learners with a VLE: firstly, how a learner subscribes to a course; secondly, how a learner starts a learning session; and thirdly, how a learner performs a learning activity. Interaction of users with a VLE builds on authentication, hence on personalised interaction and transaction.

As the use of an e-mail client as a user interface for a VLE suggests, transaction from user to VLE can be achieved with e-mails, which are sent from the user to the VLE. This mode of transaction is often used for subscription to mailing lists. For transactions with web-based information systems, e-mails are frequently used to approve the identity of a user after a transaction.

Learner subscribes to a course

In order to subscribe to a course or a learning community, a learner has to transmit information to the VLE about which course he or she wants to subscribe. This information usually is a course number (i.e. course ID). In some cases, a learner has to supply a key code for access to a particular course or learning community. In other cases, enrolment in a course has to be confirmed by a teaching person.

1. A learner finds a course or a learning community in a course directory. They have course IDs and are provided with mailto links for enrolment.
2. The learner sends an e-mail for enrolment to the VLE. He or she uses the mail account provided by the VLE, e.g. 'student.name@imap4ld.org' as a sender. The message subject has to contain the keyword 'enrol' and the course ID. This is prepared by the mailto link in the course directory.
3. The VLE sends back a confirmation e-mail.
4. The learner has to answer the confirmation e-mail. If a key code for access is requested, he or she has to enter this code in the body of the message.
5. If the learner is successfully enrolled, he or she finds a new folder for this course below the inbox folder of his or her IMAP account for the VLE.
6. In this folder, subfolders represent chapters within the course (or topics for learning communities), i.e. different units of learning. Folders labelled 'About' contain information about the course or units of learning, such as learning objectives and prerequisites.
7. One particular message folder within the course is marked to contain a recent message. This message represents the first learning activity to be performed. It is marked by the flag 'recent' and therefore marked by the e-mail client. It is also listed in the inbox.
8. A learner will select this message and find learning objects and services attached.

The following screenshot (Figure 5) provides an impression of this use case, when a learner is viewing information about the course:

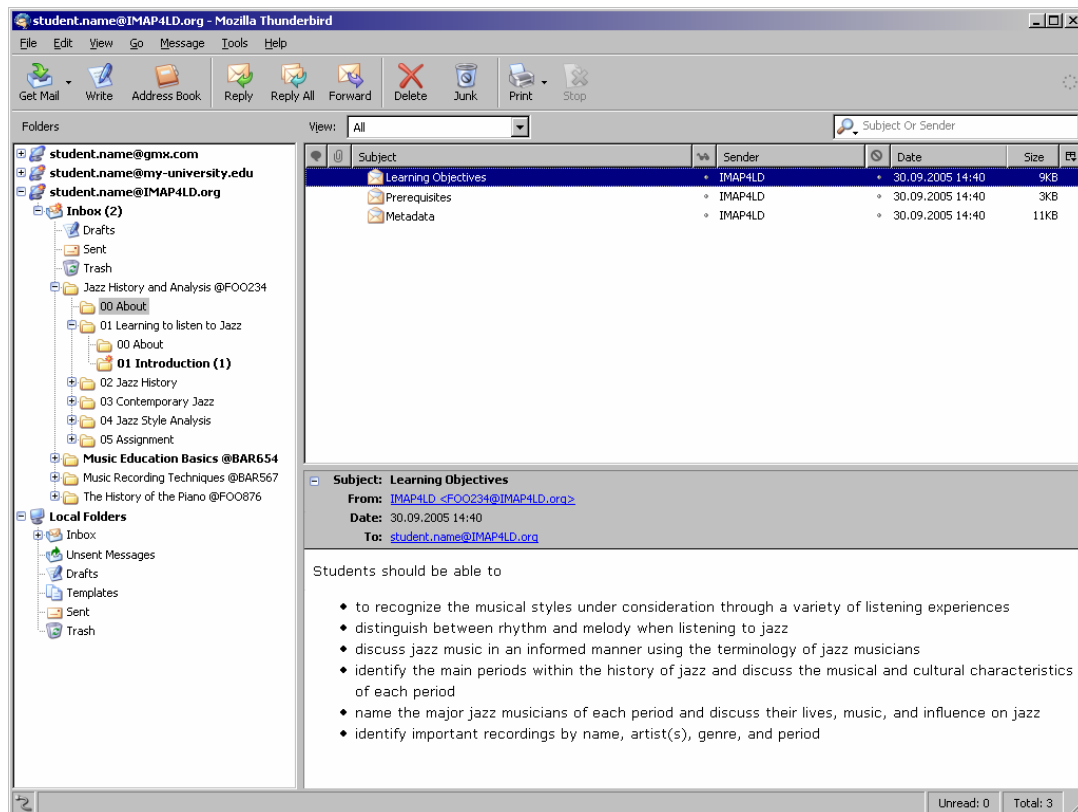


Figure 5. Screenshot for use case learner subscribes to a course

Learner starts learning session in self-directed learning

In online learning scenarios or during online learning in blended learning scenarios, a learner starts a learning session when time and space is appropriate. Any time a learner begins a learning session, he or she should attain quick orientation about the course structure as well as pending learning tasks. Often, when a learner is enrolled in different courses or learning communities, he or she has to decide about different learning tasks.

1. A learner starts a learning session by starting the e-mail client, connecting to the internet (if not already connected) and selecting his or her IMAP account for the VLE.
2. By selecting the inbox, the learner sees a list of pending learning activities. Other current news items are listed in the inbox as well.
3. The learner can decide about the sequence of pending tasks.
4. The learner can mark learning activities using the labels of the e-mail client.
5. The learner can perform a learning activity directly in the inbox or after navigating in the corresponding folder within the course, since learning objects and services needed are attached to the learning activity.

Learner performs learning activity

A learning activity is initiated by an activity description. A learner performs a learning activity directed to a sub-goal of the whole unit of learning using resources for learning, e.g. educational media, interactive tools or communication devices. Starting from an activity description, resources for learning preferably are within reach and can be used in parallel. After finishing an activity, the learner may notify the VLE of the completion and will proceed to the next learning activity.

1. A learner selects a learning activity the same way he or she might read an e-mail.
2. The activity description is given as the body of the message. Resources for learning are available directly as attachments or are linked through URLs.
3. After selecting and starting resources for learning, they either open in appropriate applications or within the web browser. The e-mail client is kept as a central navigation device for monitoring the teaching-learning process.
4. After finishing a learning activity, the learner sends a message in reply to the VLE as a notification of completion. Using the header information 'In-reply-to:' the VLE records the completion of this single learning activity.

The first screenshot of this paper (Error! Reference source not found.) depicted a learning activity presented within an e-mail client.

3.2 Possible Implementation: An IMAP Interface for LMS

The notion of having an e-mail client as a user interface for learners and teaching staff in learning scenarios interacting with a VLE is based on the idea of using IMAP and related specifications for client-to-host communication. It is not the intent of this paper to suggest using an IMAP server instead of a learning management system. On the contrary, an interface for a learning management system has to be implemented that operates instead of an IMAP server. This server has to accept IMAP requests from the e-mail clients used by learners and teaching staff on the basis of resources and services operated by a learning management system. Hence, instead of having a presentation layer in a learning management system that provides a user interface based on HTTP and HTML, a new IMAP interface has to be implemented based on IMAP, internet messaging and MIME.

Usually IMAP servers consist of an IMAP Demon *imapd*. This IMAP Demon handles requests from e-mail clients by passing them to the IMAP Mail Access Agent (i.e. the server-side part of the Mail User Agent, see Mullet & Mullet, 2000). This IMAP Mail Access Agent has access to the mailstore, which either is implemented as a file system or database (see Mullet & Mullet, 2000). In order to use IMAP for client-to-host communication accessing a VLE, the learning management system corresponds to the mailstore. A new presentation layer for this learning management system matches up with the IMAP Mail Access Agent. Hence, requests from the e-mail client used to access a VLE would be handled by the IMAP demon and passed to the presentation layer which serves as the interface for the learning management system, as shown in Figure 6 below:

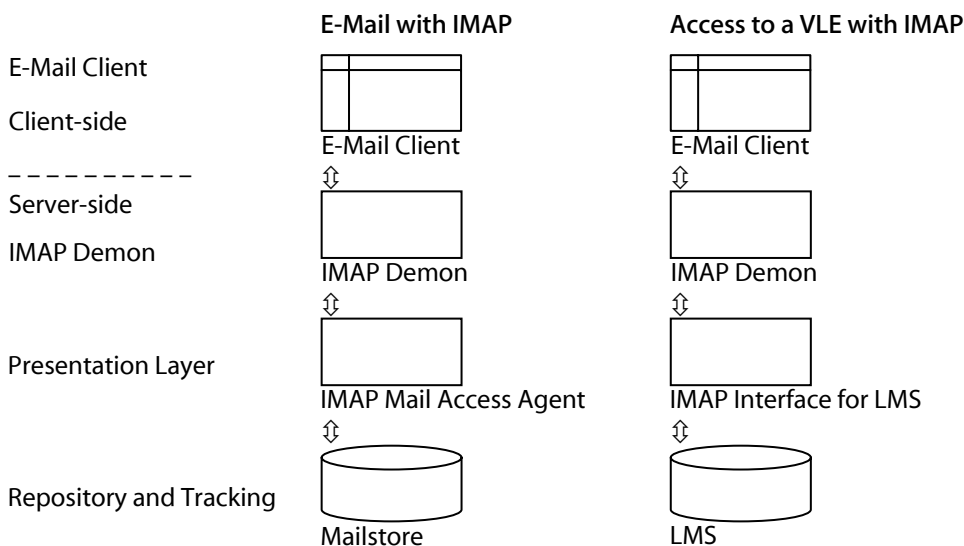


Figure 6. IMAP interface for LMS compared to IMAP server

3.3 Resulting Advantages and Disadvantages

Some of the resulting advantages were already briefly mentioned at the beginning of this paper. At this point of the examination, they can be further elaborated and completed.

Expected advantages can be summarised as follows:

- Since advanced users of digital equipment use e-mail clients frequently for communication and the organization of information, the use of an e-mail client for access to a VLE offers the potential of a **tight integration of learning in the daily workspace**. As a consequence, learning with educational media and work based on information systems becomes increasingly more integrated. Persons using a personal computer, a notebook or a PDA for work have instant access to courses and learning communities. There is no need to log on to a remote VLE by starting a web browser. Once an account for the VLE is set up in the e-mail client, the learning space is always nearby. This offers new prospects for corporate learning and may foster learning with educational media in higher education in both formal and informal settings.
- An e-mail client offers various functions for **structured access to information** available within a VLE. Thus, it allows effortless comprehension of course outlines as well as

consistent course navigation. Hence, the graphical user interface of the VLE is not of distinct visual design, probably differing from one course to another. The GUI appears to be conformant with the familiar desktop application and regular corporate communication. While interacting with services and tools and viewing educational content, the e-mail client can be kept as a central navigation device in order to monitor the process of teaching and learning.

- As a rich and native software application, an e-mail client comprises a **wide range of functions** that have to rebuild using a web browser as a client for a VLE. Authentication and sessions are established with IMAP, and synchronisation features can be used. With S/MIME even secure communication is possible.

In contrast, some possible disadvantages must be discussed as well. These difficulties are to be examined as follows:

- As outlined in the mapping between concepts of IMAP including related specifications and access to a VLE, authentication and sessions established by IMAP apply to course enrolment, course outline, learning activities and resources for learning, as far as resources are provided within internet messages as attachments or inline. However, complex learning objects, autonomous interactive tools and advanced communication services are still used in a web browser and have to be address by means of URLs. **Authentication and personalised access for some resources still have to use HTTP/HTML** and parameters in URLs or cookies. Even if the initial link to these resources is provided by a URL with personalised parameters in a message within the e-mail client, tracking of users' interaction still has to be achieved with HTTP/HTML.
- IMAP is both a line oriented and stateful protocol consisting of a small set of commands and responses. At this point of elaboration for the idea proposed in this paper, it is not possible to estimate the complexity of implementation and the resulting load for a server hosting a VLE with an interface for IMAP. A stateful protocol with e-mail clients logging on and off frequently in their online mode may put a **heavy load on a server** for a VLE.
- The idea proposed in this paper appreciates the given implementation of a wide range of functions for structuring and organising information in rich and native e-mail clients. Hence, the concept **relies on accurate implementation of IMAP** and related specifications for internet messages in **common e-mail clients** for different platforms and devices. Nevertheless, although IMAP and MIME are widespread and defined in detail, their implementation in various e-mail clients may be insufficient. This will lead to ambiguities as well as to high effort for indispensable solutions.
- Although an e-mail client is cooperative software that connects people, it is no groupware. Hence using just an e-mail client in the suggested way, a learner experiences only low awareness of other learners currently online within the VLE. It is important to point out a **lack of group awareness** for collaborative learning scenarios. A feature called 'who is online' could be implemented with messages representing other users online in a special folder below the inbox folder of the IMAP account for the VLE.

Only further elaboration and an attempt to implement the idea proposed in this paper will lead to further evidence on achievements and challenges concerning the use of IMAP for access to a VLE. Considering the implementation as an IMAP Interface for a learning management system, an existing Open-Source LMS could be extended within a particular field of application such as a university, an open learning scenario or a corporation.

Conclusion

In order to access a virtual learning environment during the process of teaching and learning, learners as well as teaching staff usually employ a web-based graphical user interface. Therefore, client-to-host communication between a user's device and the server hosting a learning management system is generally built on HTTP and HTML. As suggested in this paper, client-to-host communication can be built using IMAP and related specifications for internet messages. Some implications for the notion of having an e-mail client for access to a VLE through IMAP were examined in this paper. A possible implementation will give more insights on the suitability of this concept.

The paper asserts that common concepts of VLEs can be mapped to the use of a modern e-mail client as a well-known user interface for learners and teaching staff during the running of a course. From this concept, achieved by means of IMAP as protocol and related specifications for message content, the notion of tight integration between educational media and work

based on information systems was raised. This integration of learning within the daily digital workspace regardless of devices and platforms applies to corporate learning as well as to higher education and life-long learning, and may even apply to learning in schools. Hence, VLEs are no longer to be seen as isolated information systems accessible by means of a web browser, but can be integrated in portals, desktop applications and groupware. The concept of integrating a VLE in an e-mail client is just a first approach. Using IMAP and related specifications, this approach could be realized by the OS-community.

References

- Britain, S., & Liber, O. (1999). *A Framework for Pedagogical Evaluation of Virtual Learning Environments*. Retrieved 25.09.2005, from http://www.jisc.ac.uk/uploaded_documents/jtap-041.doc
- Crispin, M. (2003). *RFC 3501 - INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1*. Retrieved 04.10.2005, from <http://www.ietf.org/rfc/rfc3501.txt>
- Freed, N., & Borenstein, N. (2001a). *RFC 2045 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. Retrieved 04.10.2005, from <http://www.ietf.org/rfc/rfc2045.txt>
- Freed, N., & Borenstein, N. (2001b). *RFC 2049 - Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples*. Retrieved 04.10.2005, from <http://www.ietf.org/rfc/rfc2049.txt>
- Koper, R. (2001). *Modelling units of study from a pedagogical perspective. The pedagogical meta-model behind EML*. Retrieved 15.04.2005, from <http://hdl.handle.net/1820/36>
- Koper, R., & Olivier, B. (2004). Representing the learning design of units of learning. *Educational Technology & Society*, 3 (7), 97-111.
- Koper, R., Olivier, B., & Anderson, T. (2003). *IMS Learning Design Information Model*. Retrieved 17.02.2003, from http://www.imsglobal.org/learningdesign/ldv1p0/imslid_infov1p0.html
- Mullet, D., & Mullet, K. (2000). *Managing IMAP* (1. ed.). Beijing [et al.]: O'Reilly.
- Resnick, P. (2001). *RFC 2822 - Internet Message Format*. Retrieved 04.10.2005, 2005, from <http://www.ietf.org/rfc/rfc2822.txt>
- Tattersall, C., & Burgos, D. (2005). *Learning to listen to Jazz. IMS Learning Design Level B. Show case using properties, conditions, visibility and adaptive learning*. Retrieved 04.10.2005, from <http://hdl.handle.net/1820/371>

KEILab (Knowledge Exchange Initiative Laboratory): a research project about open source and elearning

Mario Vercellotti , Eleonora Pantò , Matteo De Simone
CSP Innovation in ICT, Italy
mario.vercellotti@csp.it, eleonora.panto@csp.it matteo.desimone@csp.it

Abstract

KEI Lab Knowledge Exchange Initiative Laboratory (www.keilab.it) deals with learning technologies, with particular reference to the "knowledge communities" and how they can contribute to modify education, through the free circulation of knowledge. The activities of KEI Lab articulate according to the following lines:

- *knowledge management and its diffusion in open formats, also through the use of "knowledge communities"*
- *planning and deployment of online education, both from a technological and a methodological point of view*
- *use of Open source and Free Software. The laboratory's peculiarity is to carry out the above lines integrating technological and methodological competences, through studies and researches and implementing innovative services. In this paper we present the results of the first year activities, in particular dealing with the in-house methodology developed and employed to produce a comparative analysis of about 60 free and open source Virtual Learning Environments.*

1. Introduction

KEILab Knowledge Exchange Initiative Laboratory (www.keilab.it) is funded and composed by three founding institutions:

1. CSP - Innovation in the ICT is a not profit Consortium and research laboratory appointed by the Italian Ministry of University, Research and Education (MIUR) and it represents a reference centre for technological innovation and local economic strengthening through the use of ICT. <http://www.csp.it>
2. LIASES - Laboratory of Computer sciences Applied to Economic and Social Sciences "Giorgio Rota" it is a centre of computer services that operates in the Turin University. It has itself developed an elearning environment called KLIPS, based on Zope / Plone technology.
3. DISEF - Department of Education and Teaching sciences - University of Turin; It manages a web portal about e-learning (www.far.unito.it) that organizes over 10,000 Internet resources.

The Laboratory started its activity in September 2004 and it defined an activity plan, articulated in different activity lines:

1. Showcase open source learning technology
2. Experimentation in using free software (Zope and Plone) for eLearning
3. eLearning standards Observatory
4. Applied research on monitoring features in the Free Open Source Learning Technologies
5. Applied research on knowledge communities
6. Applied research on the socio-pedagogic effects of Learning Technologies
7. Online Universities Observatory
8. Applied research about the effectiveness of the FAR (Formazione Aperta in Rete) methodology

We intend to present the results of the first year activity related to the methodology developed by the Lab to carry out a comparative analysis of about 60 free and open source Virtual Learning Environments (VLE). Such activity was aimed at realizing a deepened comparison, with a specific applied research about monitoring features. The activity is finalized to the realization of a demonstration service centre, to give assistance to people from schools, universities and small and medium enterprises in order to choose the VLE that fits best with their requirements. The Lab will also provide shortly a final report and it is developing an expert system software to help users in choosing the most suitable VLE.

2. Project objectives and operational model

In the first year of activity KEI Lab has focused on comparative analysis of free and open source VLEs.

Our work has proceeded according to these phases:

- Evaluate the current open source VLE offer;
- Deep analysis of user monitoring features of platforms;
- Define a form to collect information about VLE features that includes not only technical but also methodological issues;
- Design and prototype of an Expert System to manage the collected data;

During the first step, we have collected works produced by organizations and websites about VLE analysis and evaluation. The objective of the study was to understand the state-of-the-art in this field. To complete this activity we used several available materials: online demo areas, white-papers, websites, direct software installation on a test server, direct contact with developers and users. As a result of this survey, we have discovered as many as 60 different open source e-learning platforms. Generally for all of them their features were listed and an average judgment was assigned [1]. We have verified that the main focus of available works was set on technical aspects rather than methodological ones. These results allowed us to define the main target of our research: add methodological aspects to current analysis.

The second step stems from the consideration that current open source VLE's lack of in-depth user monitoring tools. So we produced a study analyzing open source and commercial platforms to find out about the current offer of monitoring tools and as a second step we proposed new useful features [3]. The result of this activity was useful to define the activities to carry in the third step.

The third step was a synthesis activity. Using the data gathered in the third phase, we created and formalized a form/questionnaire to evaluate the platforms. We have tried to define a methodology to collect information about VLE features that focuses on methodology rather than on technical features. So we have defined 10 macro-areas called Functional Areas to group in macro category all the available features.

The functional areas are:

1. Content management: how is it possible to create and edit contents inside the platform?
2. Standard compliance: is it possible to import/export and run SCORM packages?;
3. Asynchronous communication tools: how does the forum work? Are there other asynchronous tool?
(Announces, course calendar, internal email?)
4. Synchronous communication tools: are there chat, whiteboard, video and other tools to communicate in real-time? How do they work?
5. Personalization, flexibility: is the source code modular? Are there any available add-ons?
6. Usability, accessibility, help system: is it easy to use? Are the tools WAI compliant? XHTML compliant?
7. Language management tools: is it possible to create and manage Multilanguage contents,
is the interface available in many languages?
8. User management tools: is it possible to manage groups of users; is it possible to add users in batches?
9. Assessment tools: is it possible to create tests? Which types of questions are available? Are there any other assessment methods?
10. User monitoring tools: is it possible to track user access? Is it possible to track time permanence?

Through the results of the form [2], we associate to each VLE and each functional area a numerical value. The form is a list of questions aimed to capture a feature description. To each answer a value is linked. When the form has been filled, we sum the values obtained for each functional areas. We consider this a numerical evaluation of the VLE.

At this moment, we are writing out the forms for all 60 open source VLE we found during the first step.

The fourth step was the design of an Expert System to manage all the data gathered from the forms. An expert system is a software system with two basic components: a knowledge base and an inference engine. The system mimics a human expert reasoning process. The knowledge base is constituted by gathered data plus the analytical rules defined by KEI Laboratory. To manage the subjective and methodological aspects of this kind of analysis we decided to use the tools offered by fuzzy logic. The fuzzy logic is that kind of logic in which variables can have degrees of truth or untruth. Fuzzy logic is designed for situations where information is inexact and traditional digital on/off decisions are not possible. It divides data into vague categories such as "hot", "medium" and "cold" [4] [5].

In this case we have divided numerical value of functional areas in five fuzzy values: very low - low - medium - high - very high. This approach allows us to define rules to manage data in terms of grade of aptitude of a system to be used in collaborative, transmissive or aided (mix of transmissive and collaborative mode) e-learning courses.

3. Future Developments

In its first phase, the project has been focused on the study of the state of the art on analysis of open source VLEs, with particular attention on user monitoring tools.

For the next phase the project activities will more incisively focus on the active realization of a methodology of VLE analysis to add methodological aspects to the more common technical aspects. In order to further develop this viewpoint, the expert system will become a test platform and refinement tool for our analysis methodology.

The next problem we will face is how to choose the best VLE against an external request. We are going to complete the system defining a questionnaire to collect specific user requirements.

Besides, we are testing the different VLE's and the results of our research in real-world contexts: for example as a digital literacy initiative in public libraries in conjunction with the Municipality of Turin starting from the spring of 2005, using an open source e-learning platform.

4. References

KEI Lab documents (in Italian):

[1] M.Vercellotti - "Censimento dei censimenti e confronti di piattaforme E-learning Opensource" - 2005

http://www.keilab.it/group/keilab_site/documenti/CensiMario.pdf

[2] M.Vercellotti - "Griglia di valutazione" - 2005

http://www.keilab.it/group/keilab_site/documenti/Griglia_valutazione_final.pdf

[3] S.Basso, F.Pecchio - "Funzioni di monitoraggio" - 2005

http://www.keilab.it/group/keilab_site/documenti/Funzioni%20di%20Monitoraggio.pdf

Other documents (in English):

[4] Lotfi A. Zadeh - Fuzzy sets - Info & Ctl., vol 8, 1965, pp.338-353

[5] Lotfi A. Zadeh - Fuzzy sets - Info & Ctl., vol 12, 1968, pp.94-102

Comparing Software Development Models: Structural Problems in the Cathedral and Bazaar metaphors.

Raymond Elferink – Raycom, raymond@raycom.com

David Griffiths - Universitat Pompeu Fabra, david.Griffiths@upf.edu

Sara Zondergeld – Raycom, sara@raycom.com

Abstract

This paper builds on Levesque's identification of five issues holding back the Open Source Software (OSS) movement, and proposes that increased involvement by users in the development process can help address them. Metaphors have a valuable role in explaining to users what OSS is, and what their role involves. The dominant metaphors are the Cathedral and the Bazaar, proposed by Raymond. These metaphors have been valuable in identifying two approaches to software development. The metaphors also generate confusion, in part because of widely differing interpretations. A more structural reason for confusion is the use of vehicles which are not assigned to the same domain in the two metaphors. The negative consequences of this regarding comparison of approaches to software development are described. Adapted metaphors are proposed which facilitate comparison: the process which constitutes and maintains a Bazaar as an institution, and the process which constitutes and maintains a Shopping Mall.

1. Introduction

The authors of this paper are members of the SIGOSSEE Project Working Group on Organizational and Management Issues in Open Source Software (OSS). The project supports the use of both Free and Open Source Software in education, and is funded by the European Commission in the program 'Preparatory and innovative actions - eLearning Initiative'. Details of the project are available at www.ossite.org.

One of the issues on which we are principally focusing in our working group is the role of the end user in the software development process. It is now widely accepted that there is a role for users in the development process. We observe, however, that the majority of software development projects are conceived of as being carried out by developers only, with the end user simply acting as a consumer. Users are often involved, but principally as subjects in usability trials, and are still often seen as part of the external environment in which the software operates.

In OSS development projects, on the other hand, systems are often "custom built", with end users taking on a more complex role, acting as design consultants, providing information on requirements and feedback on the proposed solutions, or even becoming full co-designers and members of the development team. This paper is an initial step towards clarifying the role of the user in OSS development, and so to making this more effective. Our focus is on the metaphors used to describe the development process, and in particular Raymond's widely known Cathedral and Bazaar metaphors. This may not seem a central issue to those who use formal and controlled development processes, but we propose that in the OSS environment, where development is widely distributed and formal processes and control are weak, the metaphors which are used to describe the roles of the actors in the development process are of critical importance. We describe a number of problems with the dominant metaphors, examine their implications for the understanding of the role of users, and propose an alternative formulation.

2. Models of development in OSS and proprietary software

The main differences between the models of development used in OSS and proprietary software are in the areas of user participation and the degree of formality. Yamauchi [1] describes the Open Source development model as an informal development process. It is

mainly collaborative, carried out by volunteers, and communication is asynchronous and decentralized, mainly based on e-mail, web fora and chat. These characteristics contrast with commercial development models. In that context the developers are employees, the development model is fixed and the communication is often centralised and synchronous, facilitated by meetings and informal chats in the office. According to O'Reilly [2], integration of users as co-developers is a distinguishing characteristic of Open Source development models. He argues that by integrating the user into the development process, the software can be conceived of as a service, which evolves in response to the real needs of the users and not as a particular solution to a concrete problem.

3. Addressing constraints on the OSS model of development

Levesque [3], identifies five fundamental issues holding back the OSS movement, which lead to most of the public feeling that OSS is not accessible to them.

The five issues Levesque distinguishes are:

1. User interface design: the user interface of OSS projects is most of the time neglected, it is often not considered by the programmer as the `real` work;
2. Documentation: most of the time there is no documentation, there is no contractual responsibility. Mostly through bulletin boards or chat logs it is possible to get, often non-documented, answers to questions;
3. Feature-centric development: in many OSS projects, the fun stuff, the features, get more attention than the fundamental, mundane work;
4. Programming for the self: programmers tend to build their programs in a logic which is intuitive to themselves. As a result, OSS projects are mainly targeted at other programmers, although the programmers think they are open to everyone;
5. Religious blindness: OSS programmers are often blind to what proprietary software has to teach them. Every concept, even a good one, is automatically rejected.

We propose that at least three of these issues (1,2 and 4) could be addressed by involving users more closely in the development process. For this to be achieved, developers would have to welcome users into the development team and invest time to overcome the communication gap that clearly exists between the two parties. Similarly users would have to accept their responsibility and participate actively in the development team, working together with developers to identify their needs in terms of software functionality and to ensure usability by partaking in interface and workflow design.

This present paper is the first in a series of articles and web-based discussions in which we will research how far such user inclusion in software development can be achieved, and what interventions are necessary to facilitate and structure communication within this geographically dispersed, asynchronous and interdisciplinary cooperation.

In thinking about these issues, the first obstacle we came upon was the multiplicity of views on the nature of Open Source Software, and its counterpart, proprietary software. These are strongly influenced by the metaphors chosen to describe them, and in particular by Raymond's metaphor of the Cathedral and the Bazaar[4].

3. Why focus on the Cathedral and the Bazaar?

If OSS is to be more widely adopted, both in education and in other markets, users need to be helped to understand what it is, and how their role is different from that of the user of proprietary software. In introducing these unfamiliar concepts metaphors are very valuable. In 1997 Eric S. Raymond wrote an essay for the Linux Kongress, entitled The Cathedral and the Bazaar, which later grew into a book. It has been enormously stimulating for many people who are thinking about the issues surrounding Open Source Software, including the authors of this paper. The metaphor which he introduced in the title refers to different ways of creating software, and it has shaped much of the discussion of Open Source Software production. Raymond's work has also been subject to extensive interpretation and criticism, not all of which has been equally constructive and well founded. Indeed it seems that the central metaphors have taken on a life of their own, leading to a range of mutually exclusive interpretations and opinions which were never envisioned by the author. Many alternative formulations have been offered, but few are analyzed in any depth. Because of this, it is worth examining the structure and implications of the metaphors more closely, and assessing degree to which they help elucidate the nature of the Open Source Software community. This task is undertaken by this paper.

4. How the metaphors were intended

In 'The Cathedral and the Bazaar' Raymond describes two methods of Open Source Software development:

1. The traditional model of a finished product developed by a closed group of programmers with no beta released before its time;
2. The approach pioneered by Linus Torvald of widely delegated development of incremental improvements to software which is released as early and often as possible 'to the point of promiscuity'.

The metaphor introduced for the first method is the building of a cathedral 'carefully crafted by individual wizards or small bands of mages working in splendid isolation.' [4]. The aspects which Raymond identifies as being significant for the metaphor are its awe inspiring scale and internal coherence.

The metaphor for the second method is the activities of a 'great babbling bazaar of differing agendas and approaches' [4]. The aspects identified as significant for the metaphor are the many autonomous voices, views and goals which yet make up a single functioning entity.

These metaphors have proved to be very valuable in stimulating thought about the two approaches to software creation. Cathedral building is a good metaphor for a large software product produced by an effort coordinated by a single central planner, and a bazaar successfully suggests multiple actors working together with a high degree of autonomy. Despite this apparent clarity, there has nevertheless been a substantial degree of confusion as to what the metaphors actually mean.

5. How the metaphors are often used

People have used Raymond's metaphors in ways which are quite different from what he intended and explicitly stated in his paper. In the first place while Raymond was discussing OSS development, the Cathedral approach is often identified with proprietary software, while the bazaar approach is associated with OSS. This holds true in many cases, but other aspects of reinterpretation are more confusing.

5.1 The Cathedral metaphor

In some cases the focus is placed on the cathedral as a building or institution, rather than on the process of building a cathedral, which Raymond specifically refers to. This generates a number of associations which were not originally intended. Rather than representing a process, the Cathedral becomes associated with monolithic proprietary software applications. The cathedral as an institution also has strong value associations. For some it is an oppressive medieval institution which ensures its survival as a working system by imposing "one true view", suggesting a parallel with dominant commercial software development companies. For others the cathedral represents all that is good in the world, suggesting security, peace and moral authority.

5.2 The Bazaar metaphor

The bazaar has also been understood in a number of ways. In the first place it is seen as Raymond intended, as a social structure and its interactions. In other cases it is referred to as a flexible and adaptable architectural structure. It can also be seen as an institution, with all its rules of membership, enabling structures, roles, etc. As a building or an institution the bazaar would not refer to the software creation process but rather the software itself. In conversation it tends to slip from one role to the other without the participants being clear about how they are using the metaphor.

6. Comparing the Cathedral and the Bazaar

Raymond's original metaphors have been useful in stimulating and organizing thinking about Open Source Software, and have been widely adopted, so it may be counterproductive to propose completely new metaphors. The confusion surrounding Raymond's metaphors is, however, extremely intense. A trawl of the Web for references to the Cathedral and the Bazaar turns up a myriad of interpretations of the Cathedral, including as Christianity with a capital C

[5], as Microsoft [6], and as capitalism and Apple [7], while the bazaar has been associated with Islam [5] and Communism [8]. No doubt whatever metaphors we propose will also be subject to misinterpretation, but we believe that there are structural problems with the pair of metaphors chosen by Raymond which tend to exacerbate this inevitable misuse. According to Lakoff “the essence of metaphor is understanding and experiencing one kind of thing in terms of another” [9] p.5. As most people are not used to thinking of different approaches to software design, and do not have a clearly defined framework for this domain, Raymond’s two metaphors help them approach the subject. Each of the metaphors is effective in itself and provides insight into the phenomenon which they refer to. According to the Encyclopaedia Britannica [10] a metaphor is composed of “the tenor referring to the concept, object, or person meant, and the vehicle being the image that carries the weight of the comparison.”. The metaphors which Raymond establishes have not only been used to elucidate the tenors (approaches to software design), but also to compare them. Thus the vehicles (activities of a bazaar and cathedral building) are compared, and from the observed distinctions people hope to find insights into the differences between the two tenors (approaches to software design). For this comparison to be productive the vehicles should, like the tenors, be elements from the same domain. If this is not the case then the differences between the domains confuse and overwhelm the insights which may be gleaned from the comparison, and modify the interpretation of the individual metaphors.

As we describe above, Raymond is clear about the aspects of his vehicles which are significant for his two metaphors. The significant aspects of Cathedral building are its awe inspiring scale and internal coherence, while the significant aspects of the Bazaar are the many autonomous voices, views and goals which yet make up a single functioning entity. The common factor between these two vehicles is that they both have distinct types of social interaction, and to this extent they are comparable. However, the comparison breaks down if we try to compare other aspects of software development. For example different approaches to software development all work towards a comparable objective, i.e. an application. Raymond’s two vehicles have very different objectives. Cathedral Building results in the production of a magnificent and complete edifice, while the Bazaar provides a social environment within which sales persons can carry out their work. Thus, while the two tenors of Raymond’s metaphors are in the same domain (software production), the two vehicles as he defines them are clearly not. In order to make a productive comparison the reader needs to change the vehicles so that they belong to the same domain. In discussion the vehicles are usually interpreted as both belonging to one of two main alternative domains:

- a) Buildings and institutions: The title “The Cathedral and the Bazaar” invites readers to use as vehicles the Cathedral and the Bazaar themselves, both belonging to the domains of buildings or institutions. This domain is then compared with the domain of software products. This goes against Raymond’s explanation of the meaning of his metaphors, which is explicitly on processes.
- b) Construction processes: the reader can focus on the building of the Cathedral as a process, and compare it with the process of creating a Bazaar (rather than the social aspects of the activity of a Bazaar indicated by Raymond). In this case the domain of construction processes is compared with the domain of software production processes.

Both these interpretations have provided useful insights, but because the choice of domain is not usually made explicit, it is often difficult to interpret the many discussions which use the Cathedral and the Bazaar as a starting point. Thus we conclude that when using Raymond’s metaphors as a basis for comparison of software development processes it is important to specify to which domains the vehicles are being assigned.

7. Balancing the metaphors

Our experience of discussing software development using Raymond’s metaphors suggests that no matter how aware conscientious one is about clarifying the domain of the vehicles, the structural imbalance means that confusions creep into the discourse. Consequently we believe it would be valuable to find vehicles which permit consistent comparison of both production processes and software, without the confusion caused by shifting domains. We therefore now embark, with some trepidation, on providing an alternative to Raymond’s formulation, joining a motley crew of academics, programmers, and other denizens of the Web who have proposed alternatives and variations, such as planetaria [11], town councils [12], libraries [13], Baptist Church-Social Bazaar [5] and so forth. We do not suggest that our alternative is true, in the

sense that it provides a better representation of reality. Our hope is that it will prove to be productive, rather than predictive, and that the use of corresponding domains for the vehicles will make it more effective than Raymond's original version in terms of generating stimulating ideas and parallels.

We find that the Bazaar provides a strong metaphor, because it captures well the openness and fragmentation of the Open Source community. Our misgivings about the metaphor are that the activity of a Bazaar does not have a product, and so does not belong to the domain of construction processes. Moreover, the principal activities of a bazaar are those of salesmen. This has serious implications, because it suggests an identification between these sales people and coders, who are then seen as the sole actors in the software development process, producing and maintaining software and providing it to users. This is a gross distortion, maintaining the splendid isolation ascribed to the cathedral builders. In software design the product is not the source code, but rather what the program does. Users are not directly interested in the elegance of the source code, but rather in the degree to which the application meets their needs, and the source code is no more and no less than the means of achieving this. Accordingly Open Source Software is built by many different actors taking different roles, including users, system administrators, developers, accountants, marketeers, etc. All these actors need good outcomes, which address their own needs and goals.

8. A revised pair of metaphors

We suggest that we revise the vehicle by focusing on the bazaar not as a production process, but rather as an institution. The "great babbling Bazaar of differing agendas and approaches" that Raymond mentions is then not the Bazaar itself, but the process which constitutes and maintains the Bazaar as an institution. This not only allows discussion of the Bazaar as a vehicle for a product, but invites discussion of the constitution and maintenance of the Bazaar as a vehicle for a production process. This provides a much richer metaphor for the activities of the open source software community,

This adaptation would give us a balanced pair of metaphors in the sense that we now have two products, and two implied production processes. But this does not necessarily mean that the metaphors work well at a detailed level. In order to maximize the opportunities for generating potential insights we need to maximize the richness of the potential comparisons.

Consequently we require metaphors for software production in which both production methods create comparable products, with comparable properties and comparable functionality. The bazaar and the cathedral are quite different entities in function and characteristics, and so a comparison of their detailed structure does not generate a lot of insights into software production.

In order to facilitate productive comparisons we propose to use a Shopping Mall as a substitute for the Cathedral, because it has a comparable function and similar characteristics to a Bazaar while maintaining the associations of careful planning and grand scale of the Cathedral. In doing this we also avoid the counterproductive positive and negative associations of the Cathedral mentioned above. While many people may have a preference for shopping at a Bazaar or a Mall, the vast majority use both on one occasion or another, and recognize positive and negative aspects in both.

9. Conclusion

Raymond's metaphors of the Cathedral and the Bazaar are so well known and firmly established that it is hard to carry out the task of our working group (analyzing and discussing the role of the user in OSS software development) without reference to them. While Raymond is clear about his interpretations, others who use them have been less so, and many alternative interpretations and metaphors have been generated.

In our analysis we point out that the Cathedral and the Bazaar is not one metaphor but two, and that each metaphor is itself composed of a tenor (the concept referred to, in our case software production), and a vehicle (carrying the weight of the comparison). The metaphors as defined by Raymond are valuable, although we note that the real world institution of the Cathedral generates strong positive and negative associations in different people, which tends to distort interpretation. Confusion is, however, generated when Raymond's metaphors are used to seek insight into comparisons between approaches to software development models. This is because the two tenors identified by Raymond are from the same domain (software

production methodologies), but the vehicles he defines are not (the scale and coherence of a building, and the voices and goals of social institution).

From this analysis we draw three conclusions:

- a) much of the confusion surrounding interpretation of the metaphors is due to lack of clarity as to the domains to which the vehicles belong (e.g. is a cathedral referred to as an institution or a building project)
- b) Raymond's metaphors were intended to help identify approaches to software development. If they are extended beyond this function to facilitate comparisons between approaches, then it is important to specify to which domains the vehicles are being assigned.
- c) In order to facilitate potential insights into the comparison of approaches to software development it is valuable to use metaphors in which the vehicles are comparable items from the same domain. Those which we have adopted are the process which constitutes and maintains a Bazaar as an institution, and the process which constitutes and maintains a Shopping Mall. With these adapted metaphors we seek to give form to more structured discussions, which can refer to both software development processes and products. We also encourage a more rational discourse, with less emotional baggage carried over from the vehicles chosen.

10. References

- [1] Yutaka Yamauchi, et al. Collaboration with Lean Media: How Open-Source Software Succeeds. in Computer supported cooperative work. 2000. Philadelphia.
- [2] Tim O'Reilly. Lessons from open-source software development. in Communications of the ACM. 1999.
- [3] Michelle Levesque, Fundamental issues with open source software development. First Monday, 2004. 9(4). Available at:
http://www.firstmonday.org/issues/issue9_4/levesque/index.html
- [4] Eric S. Raymond, The Cathedral and the Bazaar. version 3 ed. 1997.
- [5] Michael Turner, When a bazaar is no longer a bazaar.
- [6] Interview with Jim Ready from Montavista. 2004.
- [7] Jeff Lewis, The Cathedral and the Bizarre, in www.macopinion.com . 2000. p. published as column under Mac Skeptic.
- [8] Nikolai Bezroukov, A Second Look at the Cathedral and Bazaar. First Monday, 1999. 4(12).
- [9] George Lakoff and Mark Johnson, Metaphors We Live By. 1980, Chicago and London: University of Chicago Press.
- [10] Encyclopedia Britannica Premium Service, Tenor and Vehicle.
- [11] April Corioso, The Cathedral, the Bazaar, and the Planetarium. 2002.
- [12] Alan Cox, Cathedrals, Bazaars and the Town Council. 1998.
- [13] Ken Krechmer. Cathedrals, Libraries and Bazaars. in the Association of Computing Machinery (ACM) Symposium on Applied Computing SAC March 10 - 13, 2002. 2002. Madrid.

Openness as an evolutionary determinant of human existence

Wim Westera
Open University of the Netherlands
wim.westera@ou.nl

Abstract

The topical pursuit of the 'openness' of software, content and other affairs surpasses the simple idea of making products available and accessible for users free of charge. Rather than financial, economical or technical arguments the open source movement strongly exhibits the moral aspects of open source, while referring to the equality of human individuals and their rights for equal opportunities and accessibility to relevant sources. By that, it opposes against established economic forces and expresses an ideological and, perhaps, revolutionary doctrine. In that respect the open source movement shows strong similarities with pressure groups, political factions and other movements that plead for fundamental change, if not revolution. This paper investigates the concept of openness from a linguistic, a historical and an existential perspective, respectively. It describes how the transfer of 'openness' from the domain of culture to the domain of technology can be explained and substantiated. It reveals the motives and implications of open technologies by referring to the fundamental dependence of humans and technology, and the evolutionary benefits of 'openness' for human existence.

1. Introduction

By the end of 2004 Philips, the well-known Dutch electronics company changed its marketing slogan. The old slogan "Let's make things better" was often ridiculed, because of its suggestion of an ashamed apology for failing developmental staff. The new slogan "Sense en simplicity" seems to display much more of the company's self-confidence.



Figure 1. Launch of the new Philips slogan

Philips argues as follows: 'Technology exists to help make our lives easier and more productive. So why is it so often such a hassle, full of complexity and frustration? At Philips, we believe that technology should be as simple as the box it comes in. It's this very simplicity that transforms a task into an opportunity, a burden into a pleasure. Simplicity can be a goal of technology. It certainly is the goal at Philips. It just makes sense.'

For consumers this statement signifies the reassuring message not to bother, worry, think or try to understand the apparatus, but only press the button. The positive part of this idea is that

it stresses technology's role in liberating humans from burdens and to provide a relief and enrichment of human existence. Such positivist notion of science and technology was first published in Bacon's futurist book "New Atlantis" (Bacon, 1626): it describes a utopian society whose well-being is entirely based on the sensible application of science and technology, including skyscrapers, refrigerators, airplanes and telephones, a long way before these actually were invented. With its slogan Philips chooses not to annoy people with the complex internal structure of the apparatus: it thus promotes a closed source technology approach, by literally concealing the apparatus' machinery. And to be fair, if you would like a cup of coffee, what would you want more than "simply and sensibly" push the button and enjoy the product's result? It is the result that counts, isn't it?



Figure 2. Philips Senseo: a revolution in enjoying coffee

Now, the case of the automatic coffee-machine is instructive in that it establishes a successful and 'sensible' argument for closed technologies. It may serve as a metaphor to assess the significance of closed source and open source technologies: is it really necessary to see what is inside the machine? It may even serve as an argument to challenge supporters of the open source movement.

In many respects, the open source movement seems to reflect an ideology rather than well-established line of thought that is based on sound scientific, or even practical evidence. Adherents are convinced that open source is necessary to achieve innovative software applications that support functional flexibility, user-led adaptability and the unhampered exchange of data. It opposes against existing business models that amplify the competition-based tendency to confidentiality, shielding and business patents. It thus opposes against established economic forces and expresses an ideological or even revolutionary doctrine to pursue a better world, a better life or a better future for mankind. Essentially, such premises do not differ essentially from articles of faith in religion, astrology, fortune telling and politics (in random order). The open source movement shows strong similarities with pressure groups, political factions, fundamentalist currents and other ideology-driven movements that plead for radical changes. It is important to note, however, that a great many of these action groups are doomed to disappear as a matter of course, as people gradually tend to abandon their revolutionary claims and conform to existing patterns. Clearly, such conclusion would be very disappointing and sobering for the open source movement.

This paper explores the potential significance and viability of the open source movement. It does so by investigating how the transfer of 'openness' from the domain of culture to the domain of technology can be explained and substantiated. To this end, we will first take up a linguistic perspective to explore the concept of openness and its connotative meanings. Next, we will present a historical overview that focuses on the genesis of modern society and the emergence of new and open technologies. Third, we will analyze the openness of technology from an existentialist perspective: we will explain the fundamental relationship of humans and technology and connect this with the open source ideology. In conclusion, we will combine

our findings to analyze the evolutionary benefits of openness and assess the chances of survival of the open source movement.

2. Ideals of the open source movement

In short, the core idea of open source is that the origins of a product (typically software) are publicly accessible in part or in whole (Wikipedia, 2005; Vries, F. de & Nadolski, N., 2004). Software that is developed under the open source license (GNU General Public License, 2005; Free Software Foundation, 2005) makes available the source code and its documentation to other users and developers. Other developers may join the open source community and improve pieces of source code, add new applications, port it to new operating systems and processor architectures or simply review or test existing products in order to achieve better quality software. Although a coordinating framework or group (Open Source Project) is necessary to gear separate activities to one another, the software development takes place in a highly autonomous and self-regulated way. Such approach creates a rapid evolutionary process, which produces software at surprisingly high speed compared to conventional software development methods. It is often claimed that open source software development amplifies innovation (Goldman & Gabriel, 2005). The self-correcting nature of the open source community is assumed to yield products that are understandable, well-documented, well-tested, modifiable, duplicatable and simply accessible. Users for their part are entitled to install the software without any license fees to be paid. Such absence of financial barriers creates the potential of large numbers of users. Also the common vendor lock-in, which denotes the inescapable dependence of a single commercial software provider, is avoided. Users will be supported by a community or may hire expertise from any company that has adopted the software. Increasingly, commercial software developers like IBM, Oracle, Ordina or Cap Gemini change their business strategies and adopt the open source model which focuses on support services rather than software licenses.

The open source considerations also apply for open standards (IMS, 2005). These concern publicly approved protocols and formats for data-exchange and data-storage. Naturally open source software is fully based on open standards. Increasingly proprietary software attempts to conform its interfaces to open standards so that the exchange of data with other systems is supported. Yet, specific constraints of the interfaces due to the applied implementation method are seldom made public. This also holds for the code of the internal routines and proprietary data formats. By analogy with open source software and open standards, open content refers to any creative work including text, graphics, pictures, audio or video that may be used, copied or distributed without charge. The so-called Creative Commons Licenses (Creative Commons, 2005) intend to stimulate and facilitate the actual use and sharing of information by avoiding the impediments in current copyright laws. Through a variety of licensing and contract schemes copyright holders are allowed to grant some of their rights to the public while retaining others. Open source, open standards and open content share the idea that proprietary claims hinder the products' accessibility. Its plea for openness emphasizes cooperation and transparency rather than competition and secrecy. It thus opposes established copyright claims of providers by suggesting a new business model.

3. A linguistic perspective: the concept of openness

As a first step to trace the origin and meaning of the ideology of 'openness' we will take up a linguistic viewpoint. The word 'openness' often creates positive associations. Shops, restaurants and museums are meant to be open, to welcome us and to satisfy our needs. Frustration arises each time we stumble across a shop that is closed, because, naturally, shops are not designed to be closed but are inextricably bound up with a fundamental and literal openness. As is the case with many linguistic utterances, the significance of the word 'openness' is determined by its connotations rather than by its literal meaning. Openness not only refers to a system's state of susceptibility for external stimuli, but it also signifies transparency, accessibility, frankness, fairness, hospitality, proximity and readiness for communication. Many connotations of openness suggest positive appreciations when applied to human behaviour or social attainments: having an 'open mind' clearly indicates a positive attitude towards new insights and perspectives; keeping 'open house' or having an 'open discussion' seems to be preferred above their closed counterparts; the 'open admittance' for students of the Open University of the Netherlands is a sympathetic gesture to anyone who aspires to enter higher education, regardless of formal qualifications. In contrast, 'closeness' is

associated with inaccessibility, secrecy, reserve, refusal, rejection, reticence, isolation, exclusion and many other concepts that indicate barriers for interaction. The concept of 'closeness' has unmistakably negative connotations.

Yet, it would be naive to label openness as being absolutely right and closeness as being absolutely wrong, because such statements would lack universal significance. Indeed, the word 'closeness' may have positive connotations as well. Private information, for instance medical files, may cause serious problems when they become subject of open publication or distribution. Keeping the files closed will be highly appreciated by the persons involved. Sometimes it is even advisable to shield patients from their own personal data, for instance in the case of minor or major medical risks: do we really want to know the statistical life expectancy that goes along with the possible diagnosis of genotypic, mortal disorder? The doctor's transparency may have adverse effects on the patient's mental well-being. Another example would be the so-called 'press embargo' which obliges journalists to close any communication about a certain news item until an agreed release moment. Paradoxically, this act of secrecy supports the functioning of open, free press, while it creates equal chances for different newsgroups to receive and investigate the information concerned. And what about the Nuclear Non-Proliferation Treaty, whose objective is to prevent the spread of nuclear weapons and weapons technology? Clearly, this is an example of closed content and still it enjoys (almost) worldwide support. A final example would be a penitentiary or a psychiatric institution. We would like to keep its doors closed, while their openness would cause great indignation for the majority of people, be it not for its prisoners or patients (figure 3).



Figure 3. Wanted

Obviously, the concept of 'openness' is loaded with subtleties that impede a straightforward assessment in terms of positive or negative meaning. It would be naive to conclude about the usefulness or viability of open source, just by some superficial linguistic associations that seem to reflect smart marketing rather than true ethical meaning. It is unbecoming to simply qualify the open source developers as the good guys and the closed source developers as the bad guys. With such absolute claims we would make the same mistake as astrologers, sect leaders or fanatic clergymen. The semiotic assessment of openness highly depends on the prevailing standards and cultural principles that act as a frame of reference for value judgments. In order to make progress we would need to identify agreed and accepted starting points for such framework. We prefer to take a different route. In the next section we take up a historical perspective to describe the emergence (and decline) of openness and open technologies in modern society.

4. A historical perspective: the rise and fall of openness

For many centuries the Bible has been the single example of open content. When the poor men could not afford a copy, clergymen were happy to read the holy texts in public on a weekly or even daily basis. Its openness, however, flourished by the simultaneous concealment of rival ideological content. Such monopolistic position reflects an ideological or even totalitarian doctrine rather than the free exchange of ideas between people. So while the

church provided open access to all the people, it did not go with the opportunity for individuals to choose. Here we come across an important feature of open content: indeed, diversity and the associated possibility of individual choice. The opportunity to choose is strongly related to the notion that human beings are highly independent, autonomous and responsible individuals that take their own decisions as for what books or newspapers to read or what faith to adhere (cf. press freedom, freedom of religion). The liberation of the Christian doctrine goes back to the Enlightenment, an intellectual movement in the 17th century and 18th century that strongly influenced the portrayal of mankind. It marked the liberation from the medieval doctrines of magic, superstition, prejudices and the fear of God by replacing it with human rationality and a scientific description and explanation of the world. Beliefs are not anymore accepted on the authority of priests, sacred texts or tradition, but only on the basis of reason. Not without great sacrifices (viz. Galileo Galilei) academic content acquired the status of open content by defying religious bans like the Index librorum prohibitorum of the Catholic Church. Ever since, a flood of new technologies became available. Although some of these had magical characteristics that were hard to understand (cf. the telescope), most technologies were simple and mechanical in kind and obvious for laymen (clockworks, steam engines, arms). These initial technologies were open technologies per se; because of their relative simplicity anyone could see how they worked and could replicate them.



Figure 4. Open technology: Galileo's pendulum clock

As science and technology progressed however, many new technological artefacts became available which internal operation is hard to understand. Computers, telephones and cars are complex technological devices that are being used extensively by users that mostly have not the slightest idea of what is inside and how it works. Even though the level of education is much higher than a few hundred years ago, technology is becoming more incomprehensible day by day, which seems to create a magical aura of medieval style. Through proprietary solutions and industry patents even technology experts stay ignorant about the devices' internal operation. The majority of users display an unrestricted pragmatism and goal-orientedness that consider technology as a mere instrumental utility, simply a practical means to arrive at an end. If we want to take the car from A to B, why should we bother what's underneath the hood? Why bother about the coffee machine's interior: it is the coffee's taste that matters! Accordingly, 'techno-illiteracy', especially computer-illiteracy, is growing (European Commission, 2005) and the people involved even seem to form an established subculture showing off its ignorance by degrading technology to a mere commodity. We seem to prefer plane consumerism rather than critical independence.

5. An existentialist perspective: humans and technology

The origins and effects of technological consumerism have been extensively investigated by Borgmann (1984), Jaspers (1931) and Heidegger (1977) and others. According to the existentialist Borgmann, technology promises a relief and enrichment of human existence. It liberates humans from burdens by making available a multitude of goods like heat, light, water, food, information, etc., without any effort whatsoever (figure 5).



Figure 5. Light as an incomprehensible commodity

It thus embodies the idea of easy consumerism. In ancient times, our ancestors needed a full day's work to find enough food, gather wood, make fire etc., while today, we dish up a ready-to-eat meal within a few minutes. Those were tough times: lighting the stove required knowledge, but also dedication, perseverance, goal-orientedness and involvement with the tools available. Today, the availability of goods is straightforward, omnipresent, easy, safe and immediate. Heat, light, information and coffee become available by simply pressing a button. What used to be an achievement has become a simple commodity, which demands no commitment, proficiency and skills acquired by effort, discipline and involvement with the world. The efforts are now taken care of by the device's machinery. In most devices the machinery, i.e. the technology, is deliberately kept out of sight. According to Borgmann, such pattern of separating the commodity from the machinery only leads to apathetic consumption, which is detached from any social or material context and which removes the involvement with the world. Blindfold, we locate and operate the switches that provide us with what we need, without wondering a single moment where this all comes from (cf. figure 5). Inspired by the negative effects of the industrial revolution, the existentialist Jaspers (1931) advocated his alienation thesis: technology creates a totally new material environment and causes human beings to become alienated from the world. Through mass production, human individuals are becoming more and more ignorant of the origin, composition or functioning of industrial products, be it food, clothes or consumer electronics. Prevailing values like economy, frugality and sustainability lose ground because of the availability of many identical and exchangeable duplicates. People are supposed to be trapped in a pattern of passively fulfilling their material needs by ever-replaceable stuff that is abundantly available (Verbeek, 2000).

While the Enlightenment marked men's liberation from medieval doctrines and the emergence of today's technology-based society, technology in turn seems to counteract this liberation by alienating humans from the world and making them completely dependent on the support of experts: without the help of doctors, lawyers, bakers, computer specialists, heating engineers and plumbers, we would pine away. According to Borgmann's devices theory, closed technologies, viz the concealment of the device's machineries, will affect human existence and, in the end, lead to the destruction of mankind. This conclusion would be a *reductio ad absurdum* for the support of open technologies.

Borgmann (1984) calls on breaking out this technological consumerism not by simply rejecting technology, but by claiming that technology should be open in kind: visible, accessible, adjustable, repairable. Borgmann's devices theory suggests restoring the relationship between the commodity and the machinery. Users of technological artefacts should be given the

opportunity to develop commitment with it. Devices should preferably be transparent and reveal the secrets of its machinery. To amplify the users' involvement, devices should also be adjustable to personal preferences. By making its machinery accessible, users are able to maintain, repair and adapt the devices. Indeed, from an existentialist view involvement is more important than availability. Borgmann suggests devices that support "focal practices", that is, activities that demand high degrees of involvement, that require discipline, perseverance, concentration and skills, that are physically and mentally challenging and are difficult to master, that provide satisfaction and pleasure, that stimulate rather than discourage our ties with the world and that serve no particular goal other than being a focal practice. Examples of focal practices would be walking (instead of taking the bus), cooking (instead of ordering a pizza), repairing an old bicycle (instead of buying a new one), collecting stamps, or any other activity that demands intrinsic involvement and hence serves our existential relationship with the world.

Now, this situation also applies to software. Software pre-eminently allows focal practices, while it provides many opportunities for user involvement: user-defined preferences, active object manipulation, real-time events, multiple representations of data, intelligent responses and participation in games and communities, among other things. The more open the software is, the more opportunities for involvement it provides. Of course it is not always the availability of source code that matters. The openness of software corresponds with the degree that users are able to access the software's functional "machinery", understand its operation and are able to adjust it to their own preferences. While in many respects software is becoming a critical factor in human existence, politics has widely adopted the idea to promote user involvement. The European Council has labelled digital literacy as one of its key objectives in order to establish a strong European identity, favourable conditions for the knowledge economy, active citizenship and social cohesion (European Council, 2000). Rather than learning operational tricks (pushing the right buttons for magical effects), digital literacy should focus on functional insight and understanding of software's machineries.

6. Synthesis

We have stated that the open source movement strongly resembles an ideology driven action group and, by analogy, we wondered about its chances for the future. To this end we have analyzed the concept of "openness" from various perspectives in order to assess its significance for survival. From a linguistic analysis we have concluded that the concept of "openness" tends to indicate positive meanings, but that it is also loaded with subtleties that impede a straightforward assessment in terms of positive or negative meaning. Naturally, just some superficial linguistic associations cannot establish the usefulness or viability of open source. Subsequently we have identified the Enlightenment as the cradle of modern society while it liberated man from magic, superstition and other medieval doctrines. It fostered human individuality and human autonomy, it created cultural diversity and it promoted open content and new open technologies. As technology became more and more complex, however, users were doomed to accept the technology-induced commodities as mere facts of life that, fortunately, relieve human existence by liberating it from heavy burdens. From an existentialist view we have noticed that the drawback of this pattern is that it tends to degrade users to apathetic consumers, who simply press a button to satisfy their needs and who are discouraged to wonder what is inside, where it originates from, how it is done and how it operates. In due course, people alienate from the world they live in, they become more and more ignorant of the origin, composition or functioning of industrial products and cannot but accept the commodities as magic agents to satisfy their needs. Such attitude is even furthered by the deliberate concealment of the device's machineries. Such closeness reflects an instrumental view on technology, which is destructive in kind, as it captures people in a pattern of passively fulfilling their material needs. Indeed, the Philips motto of "sense and simplicity" is likely to create dumb, lazy and dependent users, while it causes alienation from the world rather than user involvement.

At this stage the evolutionary effects have to be considered. According to Charles Darwin and Jean-Baptiste Lamarck, sons of the Enlightenment, survival depends on our ability to change. The Enlightenment itself can be labelled an era of change, if not revolution. It furthered openness and it proclaimed the ideology of upward development, progress and improvement of the world, encouraged by an ever-increasing knowledge, understanding and control of nature's processes. Abandoning change means stagnation, stagnation means decline and eventually extinction. While closeness of technology, viz. the concealment of the artefacts' machineries, is associated with passivity rather than change, it will affect human existence and,

in the end, lead to the destruction of mankind. Dumb, lazy and dependent humans will not be able to change and to adapt to changing conditions. The decline doesn't only concern our economy but will affect our culture as a whole. In contrast, openness is assumed to progress our ability to change and to amplify innovative power (Goldman and Gabriel, 2005). According to Borgmann (1984) technology should be open in kind by restoring the relationship between the commodity and the machinery: it should be visible for its users; it should be accessible, adjustable and possibly repairable in order to allow active user involvement. While closeness is destructive in kind by promoting inertia, openness is associated with adaptation, commitment, continuous development, growth and change. It suggests that openness is a precondition for survival. And this also holds for software.

But there is more to say to it. First, through the complexity of technology only few people will be able to understand the internal operation of a particular product. How many mobile callers will understand the mobile communication protocols and how many computer users have knowledge of their interrupt channels or processor drivers? Secondly, even if we would be able to understand all this, does it make sense at all? Cannot we just sit back and enjoy the commodities without bothering what is inside?

As for the first comment: we need to discern different types of user involvement. Westera (2005) suggests the following 4 types (table 1):

Table 1 Levels of involvement with technological devices

Type of involvement	Description
Sensory involvement	The device's machinery is visible, audible or tangible
Conceptual involvement	By revealing the machinery's functional components, it becomes clear how the device operates, even when most technologies are often too complex to be fully understood by laymen
Operational involvement	Users can practically and diversely interact with the devices, in order to develop their own unique methods and routines of use (cf. a piano)
Material involvement	Substantial accessibility to the machinery enables users to care for it, to maintain it and to carry out repairs and upgrades.

Table 2 Levels of openness for different software user types

User type	Level of understanding	Level of involvement
Consumer	Effects of pressing buttons	<i>Plain commodity:</i> Browsing, calculating the logarithm, etc.
Interested user	Functioning and roles of components	<i>Conceptual involvement:</i> Adjustments, personal profile and reconfigurations
Productive user	Detailed functioning and operation of some features	<i>Operational involvement:</i> Creative outcomes
Technical support	Detailed functioning and operation of all features	<i>Operational involvement:</i> Problem solving
System manager	Technical requirements, networking and operating system	<i>Operational involvement:</i> Installation and integration
System architect	Interfaces and functions	<i>Material involvement:</i> Integration and implementation
Software developer	Programming	<i>Material involvement:</i>

		Editing and programming
International working groups	Protocols and standards	<i>Material involvement:</i> Specification

The framework in table 2 contains an important message for the open source movement. Because software that is developed under the open source license only makes available the source code and its documentation to other users and developers, it is clear that its openness only covers the high end, specialist user levels. Consequently, the evolutionary benefits of open technologies and its chances of survival are limited, because the majority of users may still be fed up with simple closed software buttons that conceal the software's machinery and impede true user involvement. To be successful the open source developers should not only make their source codes available but should also enable various levels of user involvement. And what about the second question: does it all make sense; cannot we just sit back and enjoy the software commodities? The answer is simple: no, we cannot, unless we opt for a digital divide encompassing large groups of apathetic consumers, who switch off thinking and experience the world as an incomprehensible black box. The premises of modern society and the associated human rights make such digital divide unacceptable. Both software developers and software users have to make efforts to stimulate sufficient technology involvement. For a start, the European Community has introduced the European Computer Drivers' License (ECDL, 1996), which supplies and certifies the baseline computer knowledge that individuals would need in the digital age. Perhaps we should also consider a technology doctrine at schools, as is already fully accepted for reading, writing and calculus: an educational obligation, which provides basic insight in modern technological devices, a social obligation in order to avoid dropouts.

In sum, our analysis demonstrates that closed technologies are destructive in kind, by promoting inertia. Open technologies, however, including open source software are accessible, adjustable and possibly repairable in order to allow active user involvement. Openness enables adaptation, continuous development, growth and change and thus is a precondition for survival. Indeed, the open source movement will have high chances of survival, be it that its products should not only provide openness for software developers, but should also provoke various types of involvement for different user types. To be fair, we really would not resent anyone the ease of "simply and sensibly" pushing the button and enjoying the product's result. But we urge to recognize the patterns and effects of alienation and apathy. Occasionally, it may be wise to make coffee in the traditional way (figure 6). It will not only create pride and self-fulfilment. In the end it will taste better anyhow.



Figure 6. Open technology for making coffee

7. References

- Bacon, F. (1626), *The New Atlantis*. In: *Ideal Commonwealths*, New York: The Colonial Press, P.F. Collier & Son, internet wiretap edition, retrieved October 5, 2005 from <http://oregonstate.edu/instruct/phl302/texts/bacon/atlantis.html>
- Borgmann, A. (1984). *Technology and the Character of Contemporary Life*. Chicago/London: University of Chicago Press.
- Commission of the European Union (2005). *European E-Learning programme*, retrieved October 5, 2005 from http://europa.eu.int/comm/education/programmes/elearning/programme_en.html
- Creative Commons (2005). Homepage, retrieved October 5, 2005 from <http://creativecommons.org>
- Heidegger, M. (1977). *The question concerning technology and other essays* (trans. Lovitt, W.), New York: Harper and Row.
- IMS consortium website (2005). Homepage, retrieved October 5, 2005 from <http://www.imsglobal.org>
- Jaspers, K. (1931). *Die geistige Situation der Zeit*. Berlin: Göschen.
- Free Software Foundation (2005). Homepage, retrieved October 5, 2005 from <http://www.fsf.org>
- GNU General Public License (2005). Article, retrieved October 5, 2005 from <http://www.gnu.org/copyleft/gpl.html>
- Goldman, R. & Gabriel, R. P. (2005) *Innovation Happens Elsewhere: Open Source as Business Strategy*, San Francisco, Morgan Kaufmann Publishers.
- Vries, F. de & Nadolski, N. (2004) *Open source toepassingen in modulaire Elektronische Leeromgevingen, ELO's flexibel en op maat*, Utrecht, Stichting Digitale Universiteit from <http://hdl.handle.net/1820/269>
- Verbeek, P. (2000). *De daadkracht der dingen*. Amsterdam: Boom.
- European Council (2000). *An agenda of economic and social renewal for Europe*, Brussels, retrieved October 5, 2005 from http://europa.eu.int/growthandjobs/pdf/lisbon_en.pdf
- European Computer Driving License Foundation (1996). Homepage, retrieved October 5, 2005 from <http://www.ecdl.com>
- Westera, W. (2005). *Beyond functionality and technocracy: creating human involvement with educational technology*. *Educational Technology & Society*, 8 (1), 28-37. From http://www.ifets.info/journals/8_1/6.pdf
- Wikipedia (2005). Article, retrieved October 5, 2005 from http://en.wikipedia.org/wiki/Open_source

Collaborative Development Strategies for Open Source Involving the Users' Perspective

Stefanie Panke, Christian Kohls
Knowledge Media Research Center
s.panke@iwm-kmrc.de , c.kohls@iwm-kmrc.de

Abstract

The paper presents a collaborative development strategy for open source software which tackles the development process from multiple perspectives integrating technological, conceptional and task-oriented considerations. The development of the portal www.e-teaching.org, which involves many stakeholders with diverse characteristics and backgrounds, serves as a case study. In particular, we describe the creation of the editorial infrastructure based on the open source content management system Zope/Plone. On the one hand, we will analyze the history of the technological genesis and on the other hand we will outline the further implementation, focussing on the design of community features, e.g. a weblog component. As a theoretical framework we apply findings from the field of social informatics (Kling, 1992, 1999) as well as participatory development approaches like the STEPS model (Floyd, 1989).

1. Introduction

While Open Source provides many advantages, some drawbacks often exist, such as missing documentation, difficult organizational structures, lacking usability and a very technology-driven process (Levesque, 2004). To face those challenges technological, conceptional and user-oriented perspectives have to be equally taken into account. This is especially true in the interdisciplinary field of e-learning, where people with different backgrounds work together on the design of educational environments. The collaborative development of technological infrastructures holds a great effort but has the prospect of worthwhile results: In the ideal case the outcome is a product which is optimally adapted to the context of its use. This paper reflects the adaptations and extensions to the open source content management system Zope/Plone, which serves as a technological infrastructure for the educational web portal e-teaching.org.

The portal e-teaching.org is intended as an information resource for academic teachers who want to integrate digital media into everyday teaching. A target group oriented online information supply will help to improve the media competence of teachers at universities. The portal does not address merely one university, but can be flexibly integrated in hybrid qualification and support measures of various institutions. It aims at a close combination of its contents with advisory services. Therefore it offers specific functions which allow cooperating universities to generate a localized version of the portal. Furthermore, e-teaching.org is suitable for self-directed learning of individual teachers. For a detailed description of the concept and structure of the portal see Panke et al. (2004).

The portal's build up divides roughly into two phases:

- a) Pilot Phase 2003- 04: During a two year pilot phase, supported by the Bertelsmann/Nixdorf Foundation, a prototype was developed and tested at two universities.
- b) Consolidation Phase 2005- 06: Currently the Bundesministerium für Bildung und Forschung (Federal Ministry for Education and Research) supports the portal's completion and distribution to other universities. The main emphasis is, among others, on the two aspects of development of community and personalization functions.

In the course of the paper we will present the lessons learned from the pilot phase history and we will outline the further implementation, focusing on the design of community features.

2 Looking back: Analysis of Critical Factors for Collaborative Development

The science of Social Informatics studies social aspects of information and communication technologies. It is assumed that the introduction of software interacts with changes of organizational development. Social Informatics is an interdisciplinary field of research (Kling, 1999). According to this approach the development and usage of computer aided systems is dependent on an associated social context. In this course the research of Social Informatics studies the interrelations between social agents and their institutional and cultural background (Lamb & Johnson, 2004).

In this section we apply the web model of Rob Kling (1992) as the methodical frame for analyzing the first project phase from 2003 - 2004. This is a qualitatively oriented empirical method of analysis, which was designed for the analysis of computer aided systems in organizations – and therefore specially for field studies in this area. The reflection on the project's history in reference of our own experiences and the content analyses of the requirement specification, the timetables, the project reports and the mailing list serve as the data supply of our analyses. The methodical approach is structured along the dimensions of infrastructure, history of decisions, context and agents (see fig. 1).

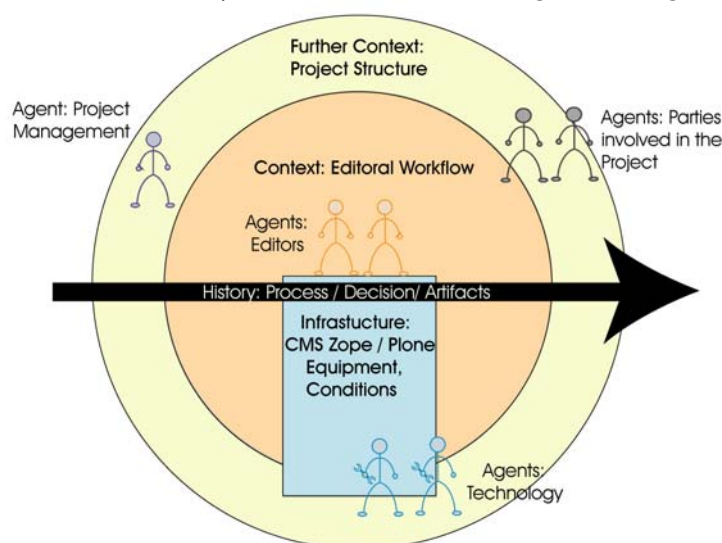


Fig. 1: Dimensions of the Web Model applied to the Project e-teaching.org

2.1 History

The historical perspective is represented by a factor of influence diagonal to the other constituents of the web model. Decisions on infrastructure and existing obligations to users and developers through the passage of time increasingly determine the developmental process and reduce the spectrum of possible realizations. In the instance of e-teaching.org the project management made central decisions at an early stage of the project: The decision on using the open source product Plone, the distribution of content and IT development to spatially allocated locations as well as the confirmations to project partners on certain functions like a local interface for each partner university. In the course of the project the importance of a close mesh of the agent groups of editing and of technology for the development of the editorial system gradually increased.

2.2 Infrastructure

The dimension of infrastructure includes the different artefacts the involved people work with; in this example in particular the IT infrastructure in form of the Content Management System (CMS) Plone, various means of communication and user- documentations for the editorial system.

IT infrastructure: The CMS Plone, used for building up the portal and for operating it, is based on the application server ZOPE and the Content Management Framework (CMF). Individual content objects like texts, pictures or files within the editorial system are not connected to the layout, but are, structured and supplied with meta information, stored as objects in a database. For doing this, Plone offers miscellaneous object types like e.g. document, picture, file, etc. Other content objects were specially defined for the requirements of the e- teaching portal, like e. g. bibliography, example of reference, personal description, glossary terms and pop-up

windows. The editorial system administrates these objects in the ZOPE object database. The different presentation formats are generated by ZOPE Page Templates.

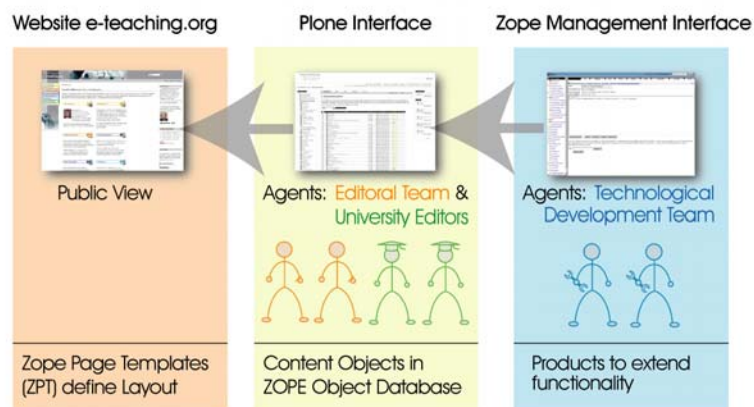


Fig. 2: ZOPE/Plone Infrastructure

As Figure 2 shows, the agents involved as editors, technicians and project partners work at different interfaces. Editors and university editors each have their own views onto the web interface, while the technicians mainly interact with the system via the ZOPE Management Interface. A "common ground", necessary for the system's development, is hampered by both, these different perspectives and the different artefacts.

Means of communication: A mailing list centrally served as a means of communication between editing, university editors and technicians. The number of 1.200 e-mails, from June 2003 to December 2004, posted on the mailing list, shows the high expenditure in coordination for the development. An important experience here shows that paths of communication – once established – are difficult to change. For a better structured communication – particularly at "debugging" – a forum, a "bug tracker", was set up after almost one year of project time. Even though this forum offered a clearer layout, it was nevertheless used actively for three months, after which there was a general return to the established form of the mailing list.

Documentation: The problem of an insufficient documentation or lack thereof is the downside of the high flexibility and adaptability of open source products. Manuals, incomplete, missing or unsuitable for the target group, are a crucial obstacle to the use of the technological infrastructure as the developer intends. Often the users are unaware of the spectrum of available functions and do not use these functions at all, or use them different from their purpose, which results in the need for further development, which in turn would not have been necessary had there been precise documentation and instruction. Mistakes in particular appeared during insufficiently documented ad-hoc decisions, which later often produced unwanted side effects.

2.3 Context

In the case of e-teaching.org the project context turned out to be a complex network of cooperation. Particularly in the beginning of the technological development the Bertelsmann/Nixdorf Foundation, as the client, was involved in decision-making processes concerning the infrastructure. In addition the editors of the cooperating universities participated in the further development of the editing system. The internal workflow of editing played a special role in the further development of the editorial system. If – as it is the case with e-teaching.org – several authors work on the same web site the setting of areas of responsibility and competence has to be ascertained. Hence a CMS usually contains a privilege management and a predetermined course of publication called a "workflow". Often a great degree of complexity is demanded for workflow support systems, in order to be able to distinctly deal out privileges. Fink, Janneck and Oberquelle (2004) however vote for an open, transparent and comprehensible privilege management. The example also yielded that formalized and restrictive workflows do not account for the actual working context of a comparatively small and compact editing team. The requirement specification – basis for the task assignment of technological development – dictated a very complex and hierarchical workflow. This document reflected the working context and the situation of interest of the project management, for example the need for quality control of the content through strict hierarchies and responsibility for contents on the management level. The predefined publication

processes of the CMS Plone correspond to the highly formalized processes in large editing teams. The following example illustrates the conceptual conflicts based on divergent models:

Example 1: Workflow

Technician 2: If- as it is currently the case- all e-teaching people have manager's privileges, then workflows are of no use- very true indeed. But this is a state of affairs a CMS is not made for. Before e-teaching I participated in the development of a system for administrating and producing encyclopaedias. The whole thing with Plone. Meanwhile they've got 340 subeditors, 40 editors and an entirety of two managers.

Editor 1: Hmm, this strikes me as odd, because we are actually working with it [the workflow]. Only then would I prefer a more detailed system if there was a problem with this model. One doesn't have to technologically solve anything which doesn't require a solution. Because of this I wouldn't like to adapt the everyday work of the people here to a workflow which I am certain works very well with 340 colleagues.

From the infrastructural framework at first resulted an unsuitable workflow which was gradually adapted to the editing process; this however with varying success. The editors had already gotten used to applying the editing system more like a publication system so that the technical representation of the multiple stage publication process was largely unused. The integration of the workflow into editorial everyday work was further hampered by technical problems and a lack and inaccuracy of documentation.

2.4 Agents

System development is an interdisciplinary project which only succeeds if there is frequent exchange of information between its agents: The technicians have to know about the work processes in editing and the editors have to have a basic knowledge of the technological frame. When analyzing this dimension, one question presented itself in particular: the question of the interaction of different subject areas. What does the interdisciplinary interaction look like in practice? In the example the agents implicitly went for different goals. While the project partners in Bielefeld saw themselves as members of the ZOPE community and where committed to the further development of the open source infrastructure, the colleagues in Tübingen identified themselves mainly with contents and the overall concept of the portal e-teaching.org. The different specialist background also had an impact on the styles of communication and work: Whereas "debugging" belongs to the everyday work of technicians, the editors' emphasis was on quality control of published contents. Accordingly mistakes where perceived as differently important. Because of unclear descriptions of occurring problems in the editorial system additional loops developed. It could not be expected, of project partners of associated universities in particular, to precisely describe technical difficulties (see example 2).

Example 2:

*University Editor 1: Mornin'[...], we've got one to 15 little problems with our internet pages ;-)
Since the change to Plone 2 (or so we believe) neither the navigational relations of our pages are correct, nor is the entire content available.*

*Technician 2: For all intents and purposes, I cannot follow you. Please be aware that such reports have to be assigned with a *comprehensive* description and possibly a meaningful screenshot! I, as a technology type, can't deal with a statement like "we believe the navigational relations are incorrect". It can be anything.*

2.5. Lessons Learned

In the example it turned out that the different models of technology, held by the participating agents, are dependent on the subject, the location and on their integration into work groups or institutions. Misunderstandings and conflicts occur when these ideas are assumed implicitly. Therefore it is important to conduct negotiations and discussions to fundamental ideas. The development of ICT did not follow a linear route from specification to the end product, but did rather correspond to a cyclical prototyping as for example described by the STEPS model by Floyd et al. (1989). This model views software design as a learning process shared by developers and users which necessitates an intensive communication. Developmental models like the STEPS model can be used to impart a cooperative style of construction in the entire project team and to create awareness of what stage of maturity and development the software product has arrived at. This allows counteracting unrealistic expectations of quality when the prototype is applied by the users. In addition the participation of the users is upgraded and requested by such a model. Experiences from similar projects show that the STEPS model can successfully guide groupware development processes

(see Kahler, 1996). However, particular design contexts may require adjustments. Pankoke-Babatz et al. (2001) provide an adopted version of the STEPS model to support the design of groupware. Whereas the STEPS design cycle foresees a redesign after a phase of software deployment in the users' working sphere the process model of Pankoke-Babatz et al. point out that this procedure is highly resource intensive and should be substituted by an integrated customization during the usage and testing phase. Especially for the design of groupware, an intense interplay of "on-road-testing" and production process is necessary. Figure 3 shows a revised version of the STEPS model which takes the specific context of content management environments into account.

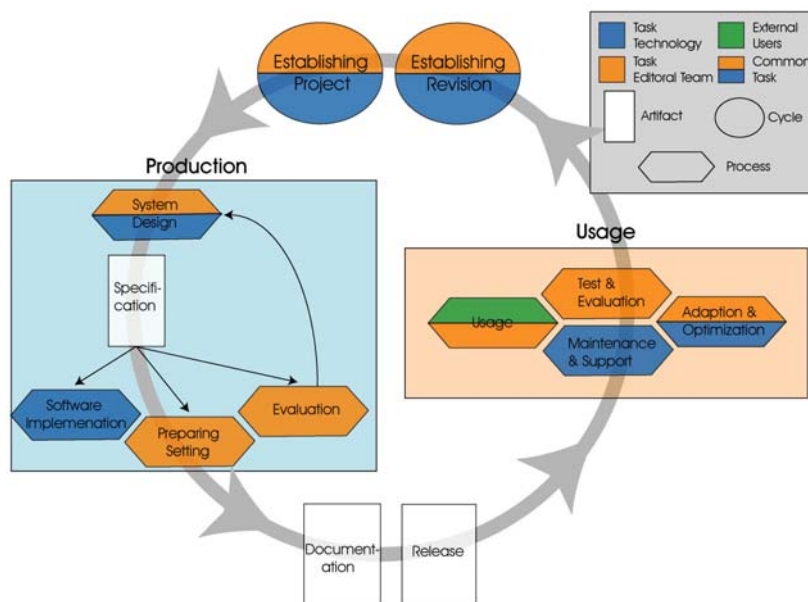


Fig.3: Software Development Cycle within e-teaching.org

3. Looking Forward: Conception and Implementation of Community Features

During the consolidation phase all dimensions of the project have evolved. Having a running technical infrastructure the focus shifted to improving usability and developing new community-oriented features. Forms of communications among editors and the technical staff changed heavily as a new job for a technician was created at the same site. Intermediate steps and partial results are discussed in weekly editorial meetings. In order to ensure transparency in decision making processes and the comprehensibility of arguments at later stages, all developmental steps are documented in a weblog, which replaced the not transparent and complex mailing list.

The context of the project has changed in a way that the project team was assigned to develop a business model which ensures the sustainable operation of the portal without third party money. While the team was granted with more independence in decision making a strong demand for market orientation arose. To bind users stronger to the portal's information and qualification services the aim was to create a community. Getting access to the community was understood as another added value for which portal users would pay.

Community members are a new type of agents and a diverse type indeed. Some community members belong to partner universities and want to find localized information about their own university. Other members act independently with different interests. As the acquisition of new partner universities was intensified, the interests of university editors weighed heavier.

To reflect upon the changes in the development process, we present the development of an educational blog for Plone. The blog is part of a larger community package which we implement in cooperation with experts and users.

3.1 Community Features

To start a community in a virtual space one has to figure out which functions and features can serve as an effective communication platform for the members. Within the project e-teaching.org a formative evaluation accompanies each phase of the development process. To inform the design of community features, both qualitative and quantitative data about the typical usage of the portal have been analyzed. This data was used to model “personas” - prototypical, fictional users (on the personas approach in software development see e.g. Cooper, 1999, Pruitt & Grudin, 2002). The personas’ needs and behaviours were matched against several community services (see Arnold et al., 2005). For example synchronous community functions such as live chat or shared workspaces were filtered because those did not match the personas’ profile. On the other hand we found that virtual business cards could help to present one’s competences and to find partners for cooperation or identify members that participate at the same scientific conferences. To establish contacts between users and experts a message system was planned. To support social networking members can publish their contact lists to others and thereby give access to their personal contacts. Social bookmarking, too, was found helpful since many users work as academic advisors and can use link lists to generate guided tours through the hypertext structure of e-teaching.org or share external knowledge resources with other members (an overview on social bookmarking is provided by Hammond et al., 2005). A weblog component combined with an RSS Feed was intended to inform frequent users about the many extensions and new articles on the portal.

Once the community services were identified, for each community function the required features were specified in more detail. First the technicians made a list of all common features found available in comparable systems. The list was matched against the requirements of the personas and discussed within the project group. In this process some possible features were detected as inadequate for the portal. Likewise some features not found in other systems but requested by the editors were added to the list. When the list was in a stable state, the technical team started to check for existing add-ons (“products”) for the ZOPE/Plone content management system. For each community service a decision had to be made whether an existing product could be customized to our needs or a new product had to be developed.

Creating new products for ZOPE is a very expeditious process. Engineering and coding the extensions is extremely efficient because the ZOPE system heavily relies on software design patterns (see Gamma et al., 1995). It provides a powerful method framework and can be extended in an object-oriented scripting language (Python) for which some good integrated development environments (IDEs) exist. These attributes allows rapid prototyping and application development. Contributing to high usability standards was one of the central aims in the project; hence features were made accessible to test users within the team at early stages to receive feedback as soon as possible.

3.2 The NotizBlog

The first implemented community service was a weblog system, the “NotizBlog”. The weblog was supposed to be integrated smoothly into the portal’s design and content structure. It provides general news about e-teaching and informs about new articles, updated texts and all activities of the portal community. New entries can be posted by the core editors of the project and by university editors. In this way partner universities can show their own projects or announce local events to a broader audience. Community members on the other hand can participate in discussions by posting comments to each entry.

After a weblog was identified as a desirable community service, the technical team started to collect features commonly available in blogs by inspecting popular blog systems. Also in the weekly held project meetings the editors were asked frequently to express additional needs for a blog system, e.g. create automatic blog entries on publishing new articles. Throughout this process it proved to be helpful that all colleagues had already gained experience in using a weblog as an internal communication tool. The list of features to be implemented was finalized in a project discussion. The technical team only stated out which features were mandatory (e.g. “Post a new entry” is a must) or expensive (e.g. a visual calendar function is hard to implement but rarely used in blogs that do not act as a diary). A program specification was created based on this feature list. By ensuring that required rather than “cool” features were chosen we avoided “programming for the self”, a phenomenon often found in open source projects.

Given the exact specification the technical team started to lookup for existing blog implementations for the ZOPE/Plone platform. At the time of the specification phase there were five implementations available for Plone. Three of the systems were in early beta states and were not likely to become publishable within the time we planned to start our blog. All available blogs relied on the user interface of Plone’s editing system. However, there are at least two user types (anonymous users and community members) who should not see the editing system. Because all blogs depended on the user interface for editors and some even

required the use of Plone folders (e.g. for hierarchical categories) none of the system fitted our needs. Plus, none of the systems provided all features requested in our specification. As the conceptional approach to integrate the blog into the portal layout differed so much from existing implementations we finally decided to create a new product. This made the user interface design more flexible, too, and allowed to design a user friendly interface which fully integrates into all views of the portal.

3.3 Documentation

One problem often mentioned about Open Source is a lack of good documentation. For the e-teaching.org 's NotizBlog both the technical and the end-user documentation were written by the technical staff. However the end-user documentation was reviewed and commented by the editors in the project team. For the technical documentation we used a third party weblog which was already in service for project communication. Each technical document was added as an attachment to a blog entry. Other technical employees could comment on the documentation. All incremental changes to the blog implementation were reported in the blog, too. This way all project members had a good overview about the development progress and were immediately informed if some features changed, e.g. how to login or where to define new categories. The documentation process became more transparent and the developers were encouraged to write documentation continuously rather than putting this sometimes unpleasant part on a to-do-list.

End-user documentation on the other hand must provide a fast introduction to the application's most common features and a reference for the less frequently used features. For the NotizBlog a written document consists of a complete description of all features and gives some background information, e.g. explaining the idea of blogs in general. A quickstart tutorial comes in the form of an annotated screen capture movie. The film shows the original web screens of the application and uses bubbles to explain each step. The user can interactively navigate from step to step and reshoot each step several times. Using such movies is a very appropriate method to train users for an application. It demonstrates instantly the most important features and motivates the users to work with the system. Even more important, the documentation is more accurate because each operation is described completely if all steps are shown. Recording the screen captures can also be in charge for usability testing as the next section will show.

3.4 Usability testing

The close cooperation between technical staff and editors led to a specification that highly appraises the user needs for both functionality and the design of the user interface. However, the actual usability performance can only be measured on a running system not on some sketches. We used three methods to detect insufficient or dysfunctional operations and to improve the interface accordingly: capture the steps to achieve a goal, observe users on their first time use of the system and regular request for feedback from test users who work with the system.

In an early stage the technical team produced screen capture videos that demonstrated the features of the blog. This forced them to review each step to fulfil a given task. The granularity achieved by splitting the task into single steps can be used to evaluate the performance according to the GOMS (Goals, Operators, Methods, Selection) model (John & Kieras, 1996). In this context it ensures that there are no redundant or ambiguous steps involved. In many cases drawbacks in user interfaces are faced by developers with workarounds. After a while the workaround is taken for granted, the developer no longer perceives it as a lack of usability. Producing the videos helps detecting unreasonable steps: The higher effort in reaching a goal is mirrored in more editing steps within the process of screen capturing. This method detects rough violation of usability even before the first running version is delivered to the test users. Figure 4 shows an example:

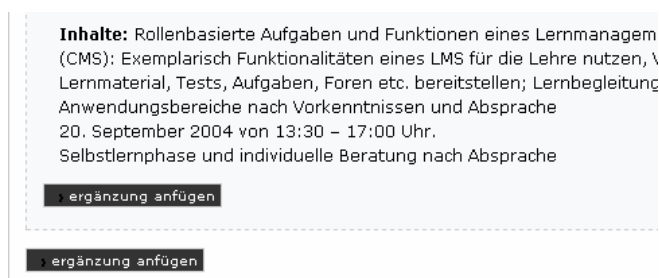


Fig. 4: Screenshot taken from an earlier version of the content management system's interface.

As one can see in figure four, the button 'ergänzung anfügen' is doubled. Only one button performs the correct operation. This was ignored by the developer because he knew which button to press and he did not perceive the wrong button any more. But when he had to annotate the screen capture and to explain where to click he recognized the ambiguousness and posted the issue in the weblog as a bug report.

For the NotizBlog the complete project group acted as test users. A first version was tested by observing some of the editors using the blog. Though this was done rather informal some options for improvement could be instantly found, e.g. to reorder menu items or re-label buttons. However, most of the usability failures show up when people work with the system in a realistic context. Thus, the project employees were asked to give regular improvement feedback. One major challenge in this phase is to motivate the users to contribute to a system only for test reasons. For this purpose the new blogging infrastructure was embedded in an informal communication setting. By coincidence most of the project members were very interested in jogging and inline-skating, so the testing implementation was used as a "start blog" where the latest running performances could be reported. Choosing a social topic for testing turned out to be a wise decision: The team soon became very active in posting their training results and resolutions. The intensive use of the blog uncovered some less obvious usability issues which could be fixed before the first release on a public server.

The usability testing of the weblog component is a good example how project communication has shifted towards a participatory development cycle: On the one hand the users were encouraged to report bugs and their participation was appreciated; on the other hand the users themselves were assuming responsibility for the debugging process. They tried out new functions and did not bother about errors. As a side effect the testing served as effective software training, too. This explorative learning strategy lead to a better understanding of the software and intensified the knowledge about features within the core editing team.

4. Conclusions

The ZOPE/Plone architecture enables efficient development of new features. However, having a large number of features is no end in itself – neither for developers, nor for end-users. A feature-centred development may be satisfying for the technicians in the beginning, but as soon as they are confronted with negative feedback from real users enthusiasm can change to frustration. Users in turn often do not appreciate a wide range of features because this only complicates the learning process. They expect a clear selection of features they really need.

In a user-centred development process end users need to be closely involved during all phases. Observing and understanding the working processes of the users is obligatory. Frequent meetings between users and developers are needed to operate on a common ground and to ensure that users and developers are aiming at the same target.

The cooperative and participatory design and testing of features within the new project architecture of e-teaching.org is mirrored in the specification and implementation of the blog component. As a result, the weblog reflects the actual needs of the users. As for the other community services of the e-teaching.org portal similar strategies are applied. In opposition to the blog other services will focus on different user groups. While the blog is mainly used by editors of the team, social networking and bookmarking functions will be used by community members from partner universities. To involve users from this group in the development process a common workshop will be held to present community features and to receive feedback and user-requests before the actual release.

Our positive experiences with involving end users more closely into the development process can be transferred to other open source projects. The time saved in rapid application development should be invested into researching users' expectations and working context. In many cases, the design of user interfaces and the production of informative documentation influences user satisfaction more effectively than coding more "add-on" functionality.

5. References

- Arnold, P., Geiser, B., Panke, S. (2005). Personas im Designprozess einer E-Teaching Community. In: J. M. Haake, U. Lucke & D. Tavangarian (eds.). *Proceedings DELFI 2005, 3. Deutsche e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)*.
- Cooper A. (1999). *The inmates are running the asylum*. Indianapolis: Sams.
- Floyd, C., Reisin, F.-M. & Schmidt, G. (1989). STEPS to Software Development with Users. In: C. Grezzi & J.A. McDermid (Eds.). *ESEC '89, Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, pp. 48-64.
- Fink, M. Janneck, M. & Oberquelle, H. (2004). Benutzergerechte Gestaltung von CSCL Systemen. In: Pape, B., Krause, D, Oberquelle, H, (Eds.) *Wissensprojekte. Gemeinschaftliches Lernen aus didaktischer, softwaretechnischer und organisatorischer Sicht*. Münster: Waxmann. pp. 203-219.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series.
- Hammond, T., Hannay, T., Lund, B. & Scott, J. (2005). Social Bookmarking Tools (I): General Overview. In: *D-Lib Magazine*, 11(4). [Online] Available: <http://dlib.org/dlib/april05/hammond/04hammond.html> [29/09/2005]
- John, B. E. & Kieras, D. E. (1996). Using GOMS for User Interface Design and Evaluation: Which Technique? In: *ACM Trans. Comput.-Hum. Interact.* 3(4), pp. 287-319
- Kahler, H. (1996). Developing Groupware with Evolution and Participation: A Case Study. In: *Proceedings of the Participatory Design Conference 1996*, Cambridge, MA, November 1996, pp. 173-182.
- Kling, R. (1987) Defining the boundaries of computing across complex organizations. In: R. J. B. Jr. & R. A. Hirschheim (Eds.). *Critical Issues in Information Systems Research*. New York: John Wiley, pp.307-362.
- Kling, R. (1992). Behind the Terminal: The Critical Role of Computing Infrastructure In Effective Information Systems' Development and Use. In: W. Cotterman & J. Senn (Eds.). *Challenges and Strategies for Research in Systems*. New York: John Wiley, pp. 365-413.
- Kling, R. (1999). What is Social Informatics and Why Does it Matter? In: *D-Lib Magazine*, 5(1), pp. 217-232
- Lamb, R. & Johnson, S. (2004). Social Aspects of Digital Information in Perspective: introduction to a special issue. *Journal of Digital Information* 5(4).
- Levesque, M. (2004), Fundamental issues with open source software development. In: *First Monday*, 9(4), [Online] Available: http://firstmonday.org/issues/issue9_4/levesque/index.html [10/14/2005]
- Panke, S., Wedekind, J., Reinhardt, J., & Gaiser, B. (2004). *www.e-teaching.org – Qualifying Academic Teachers for the E-University*. In D. Remenyi (Ed.), *Proceedings of ECEL 2004, European Conference on e-Learning* (pp. 297-306) [CD-ROM].
- Pankoke-Babatz U., Prinz, W. Wulf, V. & Rohde M. (2001). *Spezifika des CSCW-Designs*. In: G. Schwabe, N. Streitz & R. Unland (Eds.): *CSCW Kompendium - Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten*. Berlin: Springer, pp. 373-394.
- Pruitt J. & Grudin, J. (2002). Personas: Practice and Theory. In: *Proceedings of the conference on Designing for user experiences*. San Francisco: ACM Press.
- Schulz-Schaeffer, I. (1996). Software-Entwicklung zwischen Ingenieur- und Designwissenschaft. Überzeugungskraft und nützliche Widersprüchlichkeit von Software-Engineering und Software-Gestaltung. In: H. D. Hellige (Eds.). *Technikleitbilder auf dem Prüfstand. Leitbild-Assessment aus Sicht der Informatik- und Computergeschichte*. Berlin: Edition Sigma, pp. 115-140.

E-LANE: Open Source eLearning in Latin America

Dudley Dolan
Trinity College Dublin
dudley.dolan@cs.tcd.ie
Carlos Delgado Kloos, Mario Muñoz Organero, Abelardo Pardo
Carlos III University of Madrid
{cdk, munozm, abel}@it.uc3m.es

Abstract

This paper explains the development of an open source eLearning platform within the E-LANE (Europe and Latin America New Education) Project. This project sets out to integrate applications already used in the context of eLearning as an open software eLearning platform, using the .LRN educational platform, the design of an innovative teaching methodology oriented towards this platform, and the integration of course content from several educational institutions in Europe and Latin America. The objective is to provide Latin American society with low cost educational material.

A set of courses has been produced which are divided into two categories: basic skills and life long learning. Courses for basic computer literacy have been developed according to the ECDL syllabus together with other additional courses. All the courses are packaged in SCORM, and their content was designed according to the E-LANE methodology and E-LANE pedagogical model.

1. Introduction

E-LANE (European-Latin American New Education) [3] is a project funded by the European Union with the following objectives:

- the integration, customization and extension of solid open source e-learning applications in order to build an enhanced Learning Management System (henceforth LMS),
- the definition and development of a new and fully tailored methodology for course design, development and validation,
- and the creation and dissemination of courses using this methodology and deployable in the developed LMS. These courses are deployed in several Latin-American countries.

The aim is to maximize the impact of this project both at the level of academic and non academic training and to promote a solid environment for long life learning in the future information society.

In order to achieve this goal, the project is organized into several subprojects each of them with concrete interrelated objectives. At the conclusion of the project, a fully open source e-learning platform will be available for distribution. Furthermore, a set of courses from different disciplines and levels will also be available. These courses are organized into two main categories: basic skills and life long learning.

Aside from the open source e-learning platform and the content, the project also puts emphasis in both the methodology to develop the course material, as well as the maximum widespread dissemination of the courses.

The E-LANE project is well aware of the importance of open source tools in the development of e-learning applications. Open source tools are extensible, customizable, adaptable and greatly reduce the total cost of ownership. Since the project provides LMS, content and methodology with open source content, the paper shows how all these aspects interconnect to provide a complete e-learning platform.

The rest of the paper is organized as follows. Section 2 describes the open source software platform used as LMS. Section 3 explains the type of courses designed within the project frame.

Innovative methodological aspects are discussed in Section 4. Section 5 explains the techniques used to measure the impact of the project as well as to refine the platform. Dissemination and sustainability of the entire proposed methodology is presented in section 6. Section 7 describes the specific context in which the project takes place. Section 8 provides a deeper insight in the project structure. Finally, Section 9 includes several concluding remarks.

2. Open Software Platform for eLearning

A key element for an elearning platform is the underlying technology. In today's society, elearning has the opportunity to provide teaching support for large audiences. It is precisely in this context where new pedagogical resources can be deployed. Despite the fact that several elearning platforms are slowly appearing in educational institutions, they mainly provide support for generic administrative tasks as well as some basic tools for teaching support. At the same time, in the last years there has been a very significant effort to define different standards that allow the exchange of teaching material between different platforms.

The E-LANE project has developed new capabilities to enhance an already powerful LMS based on .LRN (an open source e-learning server platform). .LRN was chosen as the starting point for further developments because .LRN is an open source package inside the open source library OpenACS based on the server AOLServer.

The platform developed on top of .LRN by the E-LANE project has a client/server architecture and imposes minimum requirements on the client side. The server side of the platform is built upon the following open source sub-systems:

- PostgreSQL Database: An open-source heavy-duty database server. Its presence has increased significantly in recent years and so its reliability [8].
- AOLServer: An industry-strength free distribution web server containing a tightly integrated TCL interpreter. The scalability and performance of this server has been proven in numerous scenarios [1].
- OpenACS: An open-source comprehensive library of web functions that allows for fast deployment of web functionality. This tool has a very active community continuously improving its content. It has a modular structure where the functionality is captured [7]. The main advantage of OpenACS over its competitors is that it is highly scalable, well integrated, and has an excellent architecture for building programs that interoperate with others.
- .LRN (read "dotlearn"): A modular package to be deployed within OpenACS that implements an e-learning environment [6]. Originally developed at MIT, .LRN can be used to support a range of applications, including course management, online communities and collaboration.

The E-LANE project not only uses the .LRN educational platform but also is contributing to the development of this platform to expand and improve its functionalities. The E-LANE project has improved the capabilities of several sub-systems thanks to the implementation of a series of additional packages which are designed, implemented and tested to be fully compatible with them. The main developments already finished so far are: Random Photo Portlet, E-LANE customization, Evaluation package (GradeBook), User Tracking package, Knoppix CD, inclusion of the IMS QTI [5] standard for the assessment package, Mail-handler package, Modifications for the LORs package, and Modifications for the forum package.

The project is currently working on additional developments like the inclusion of the IMS Learning Design standard [5], web services, capabilities for moving objects between courses, a Word to DocBook converter, new functionalities for LORs and new editor interface for assessments.

A test server is also available. Through this server one can test whichever .LRN functionality and if there is any bug one can send a report for trying to solve it and including a new test.

Releases of the E-LANE Project distribution are provided based on the latest stable release of the .LRN platform. The E-LANE distribution is a superset of .LRN: it contains ".LRN-certified" components according to the tests. The latest E-LANE release is publicly available.

3. Course Content

A concrete set of courses will be produced in order to be taught with the proposed tool and methodology. By combining the effort of the partners, already proven course material will be made available to implement the appropriate courses that address the educational needs of the Latin American societies. The fact that the consortium consists of five Latin American partners from five different countries provides the ideal scenario to create content with a lot of variety and test it in a wide variety of situations ranging from conventional university environments to communities with special needs. The courses provided by the project are divided in two categories: basic skills and life long learning.

The first category includes courses oriented towards teaching users basic skills in a variety of contexts. Perhaps the most significant is an effort to convey basic computer skills to a large portion of society. Initiatives such as the International Computer Driver License (ECDL.com) [4] are crucial to define the set of basic skills required to bring the information society closer to the citizens.

The second category includes courses oriented to enhance the learning experience of people by completing their education with additional courses. From this point of view, the project consortium is participated by Latin American educational institutions with a large experience in this context. The generated content also takes into account the following initiatives:

- Creative Commons [9] that offers a flexible range of protections and freedoms for authors and artists
- The Open Sustainable Learning Opportunity (OSLO) Group [10] that carries out research, development, and implementation initiatives that strive to make educational opportunity freely available to all who desire it.

4. Innovative Methodology

Equally important to the software platform is an innovative methodology that capitalizes in the new resources provided by technology. More precisely, both the way in which courses are taught, as well as the way material is prepared needs to be taken into account if it is to be used in the context of elearning. From this point of view, the consortium counts on the participation of experienced partners with a proven history of successful projects that will deploy these new methodologies and resources in the context of the provided platform. The methodologies take into account the scenario in which a person needs Life Long Education.

A second aspect that is crucial to develop material is to accommodate the required infrastructure to be able to achieve a high degree of reuse. All the universities participating in the consortium are well established educational institutions, and therefore they provide a large amount of content for courses that needs to be integrated to achieve maximum impact when deployed. As result of this methodology a new paradigm for education in Latin America will be produced. This paradigm will take into account students with different needs in order to maximize the social impact of the courses.

As an example of this methodology, the project is currently devoting a significant effort toward including support for the IMS Learning Design specification. This proposal provides a framework to use a wide range of pedagogies in on-line learning by means of a flexible and generic language. The specification is based on the definition of activities, roles and the flow that controls how the activities are performed by the different actors according to their role. The specification is divided in three levels (A, B and C), each level adding some functions on the previous one. Level A defines the language to specify activities, roles and how the different activities are presented to the users depending on their role. Level B introduces properties and conditions enhancing the capabilities to define conditional activity flows at design time. Level C introduces notifications that can be trigger in several situations such as one an activity is completed or when a property changes its value.

5. Impact Measurement and Refinement

Social impact is of the utmost importance. The objective of the project is to maximize such impact, therefore, a carefully planned evaluation stage is essential. In this phase, all the required methods will be made available to gauge with as much precision as possible which parts of society most benefit from the courses, if they have been efficiently implemented, the required resources, their contribution to improve the surrounding environment, etc. Derived from this detailed study, an improved platform is proposed in order to correct and improve all the required aspects. This phase is also very important, since the platform needs to be developed by taking into account as many environmental aspects as possible, but these aspects cannot be experienced until the platform is put to a test and its functionality submitted to a regular use.

6. Dissemination and Sustainability

This objective is oriented to extend the impact of the platform, not only to the students that receive education with it, but also to the entire educational and scientific community. The fact that the platform will be distributed with an open software license will significantly contribute to its dissemination. Other technology areas have been greatly benefited by tools distributed with such licenses. Elearning should not be an exception. In fact, this dissemination is expected to grow when more and more users and institutions are capable of exchanging solid and effective course material.

As for sustainability, the objective is to get involved in the project different institutions such as governments, trade unions, community organizations, etc. These organizations already have needs in terms of training. Governments in Latin America are already contributing to provide widespread education to a wider range of the population. This project is a perfect vehicle to improve or complement these projects. Trade unions are very concern with the increasingly important area of on-the-job training. Workers can substantially improve their skills by using a flexible platform that adapts to their learning environment.

7. Latin America and Europe: A worthy Alliance

If there is a concept fully fashionable nowadays, that's the "Information Society" one. And it is not just a question of caprice, a temporal tendency that'll be over in a few months from now. The information society is the seed and the engine of the New Economy, and as a consequence it becomes the factor that's changing the whole conception every human being has of the world.

This new kind of society and the new economy that constitutes its base offers a quite interesting mix of opportunity and challenge. If a revolution is about to come, or is coming right now, we cannot miss the chance to fix the inequalities derived from back in time as the Industrial Revolution. That is to say, developing countries have an opportunity to move with the First World, as a consequence of this new beginning. But any new opportunity involves new challenges: the higher the potential benefits, the more important the effort to be invested and the risks to be faced. In Chris Patten's words (ex-governor of Hong Kong): "Neither Latin American nor Europe can miss this boat." And a very convenient way to be sure we won't miss that boat is to work together.

Europe has the means, and Latin America the potential. We only lack, at the moment, a target. What is the macroeconomic aggregate that quantifies the effects of this new economy? According to the magazine "Indicadores de la SI en Iberoamérica", the ICTs market is the best indicator, added to others like the number of Internet users, the profile of these users, number of PC units, E-Commerce, etc.)

So, in order to take full advantage of this continental alliance, we need to bring to practice projects adapted to the data shown by these indicators, a major goal being to improve them.

As we will see later, Latin America is not very well situated in the charts. But the present is not the best part when talking about this geographical region, the future is. Latin American markets are growing, and they are expected to grow even more, as technological processes penetrate more and more into Latin American countries. And Europe should encourage this, should help Latin America turn into the powerful ally the Old World needs.

7.1 Two gears in Latin America

When contemplated from the outside, one tends to regard Latin America as a homogeneous set of developing countries. No wonder many initiatives to improve Latin America economy have failed. This sub-continent covers an extremely rich set of completely different realities, ranging from the most exuberant wealth to the absolute misery, from brilliant PhDs, thinkers, scientists and writers to thousands of illiterates. There are no two Latin American countries whose situation can be considered similar. But the differences go far beyond. Every community, every people and race, even every city, presents different features, a different personal bias. We are more likely to succeed if we begin by presenting a summary of the global situation. That is, an "aerial view" that will allow us to take a first look at the land where we will be walking later.

What is the global situation in Latin America, in terms of ICTs, then? Let us review some figures (source: International Telecommunication Union):

	Europe	Latin America	World Total
Number of Internet Users (millions of people)	171.3	25.3	544
Number of PC Units (millions of units)	140.6	28.2	471.1

Clearly, there is an enormous gap between the two allies. This constitutes a danger not only for the adequate development of the information society in Latin America (which in case of it not eventually developing, should be consider a major historical failure), but for the success of the alliance, also.

As remarked before, it is fairly convenient to analyse in more detail the problem, before proposing any course of action. It is the Latin America with "Two Gears".

7.2 Differences between Countries

It makes no sense to consider the situation in Brazil and in El Salvador, for instance, as similar. Oversimplification is the first thing to be avoided. According to data from Dataquest, related to the growth in the number of hosts in Latin America, there are two different gears, two completely different sets of countries, at least from the point of view of the penetration of information and computer technology (henceforth ICT): Brazil, Mexico and Argentina are leading the race, while the rest of the countries remain far behind. Let us see some figures which reinforce this idea (source: International Communication Union):

	Internet users (millions)	Hosts (thousands)	PCs (millions)	Mobile Phones (millions subscribers)	Online Spending (millions dollars)
Brazil	8	1644.6	10.8	28.7	4256
Mexico	3.5	918.3	6.9	20.1	1542
Argentina	3	465.3	2	7	1094
Colombia	1.1	57.4	1.8	3.2	336

Chile	3.1	122.7	1.3	5.3	312
Venezuela	1.3	22.6	1.3	6.5	348
Peru	3.0	13.5	1.3	1.6	164
Other countries	3.2	165.6	2.9	10.8	777
Total	26.2	3410	28.3	83.2	8829

7.3 Differences inside each Country

A simple division between “fast adopters” and “slow adopters” is not enough. Even in Brazil, the most advanced country in Latin America in terms of technological development, internal differences are enormous. This implies the need for work on technologies focused on local and regional scales, attempting the development of products and systems adapted to each area.

Applying the concept of “two gears” to the internal differences is a must if we wish Latin America to join the information society properly, keeping in mind that we should divert globalisation to be at the service of individuals, obtaining a development process that takes into account not only financial issues, but also people.

Two gears have been identified in the following internal dimension:

Rural and indigenous communities: According to the UNESCO, illiteracy rates in Latin America are really high. In Guatemala, for instance, this rate reaches 40% among the adult female population. Most of these illiterates are from rural and indigenous communities. What makes the problem even worse is that they lack the knowledge and also the means to get access to that knowledge. Some attempts have been made to reduce this impact over the years, most of them based on the use of ICTs, obtaining results both promising and successful. But even with this kind of solution, lack of resources (in this case, the almost complete absence of telecommunications infrastructures) arises as a major hindrance. And even in the cases when telecommunication infrastructure exists, bandwidth is very small. These obstacles are high, but in no way impossible to overcome. One has to adapt the solution to the given resources.

Firms, specially Small and Medium Size Companies: The situation in the business sector is completely different. Without attaining the figures of Europe or the USA, Latin American companies are not so badly prepared for the information society and the new economy. Let us take Chile as an example: 43.6% of the firms are connected to the Internet, 14% have some kind of web presence, whereas 19.5% are connected around the clock (source: Tercera de Chile Newspaper).

7.4 How E-LANE Addresses This Situation

No analysis is useful if it does not aim for the elaboration of some agenda for action or, at least, some kind of suggestion to encourage improvement. That is the final aim of our proposal: we are not only interested in describing the situation, but in contributing to the development of the information society in Latin America. One of our guidelines to do so has been presented: implementing pan regional models is in no way advisable, so, as we will detail later, we propose a system perfectly adapted to the “two gears” concept.

That is not enough. However, what is the objective of our solution?

As Bala Veeramacheneni (Chairman of Bazzare and professor in the University of New York) states, the goal is to transform technology into something useful for people and, above all, to form human resources properly, because human beings and their formation is the most important thing in the information society. Something similar can be deduced from the words by Román Mayorga, from the Banco Interamericano de Desarrollo (Venezuela), when he says that the most important problem in Latin America is neither the lack of infrastructure nor the cost of new technologies, but the lack of creativity in the use of all these resources (focusing always on local needs and priorities, of course); the lack of creativity, in a word, to use information and knowledge to satisfy their own needs. Fortunately, this creativity can be improved and even learnt.

As knowledge seems to be a key factor for the social and financial development of Latin America, formation and training arises as the way to follow. There is an old saying: “Give a person a fish and you feed her/him for a day, teach her/him to fish and you feed her/him forever”. That is our spirit.

Education makes human beings free and increments their capacity for action, as told by José Antonio Hernández, Secretary of the Spanish Agency for International Cooperation. Furthermore, education brings the chance for a wider social dialogue.

Bringing about social change by means of individuals' education sounds good. The next question is: what kind of education? This question cannot be faced in isolation. We need to establish the physical resources we plan to use. As Marshall McLuhan used to say, "nowadays medium is message". Though this sentence can be regarded as a little bit exaggerated, it suggests that, in our case, the best way to educate for the information society may be to use the resources and systems provided by this information society itself. Teaching people how to cope with the information age and computer networks by means of these same networks and computers is a sound option, definitely.

One could wonder if Latin America, globally, is ready for this kind of high technological teaching and learning processes. Mayorga, author of books like "La universidad para el cambio" and "Cerrando la brecha" ("Bridging the Gap"), considers that, even if in Latin America the use of computers for education is still emerging and very new, there are quite a few valuable experiences.

Summing up, these are the main components in the E-LANE project, all of them derived from and based on ideas previously presented:

Two gears: low-speed connectivity and high-speed connectivity. We intend to reach a wide range of target populations.

Two dimensions: rural and indigenous communities (providing them with literacy and computer literacy), and Small and Medium Size Enterprises (henceforth SMEs), being our intention to cover such a high number of social sectors as possible.

Practice oriented: Our aim is to set up a demonstrator of courses and contents, delivered through a sophisticated and innovative tool with monitoring of performance of the students.

8 The E-LANE Project

The main structure of the project is organized so as to maximize its social impact. Both the content as well as the partners in the consortium provide the ideal scenario with which to bring affordable educational technology and content to a large portion of the population in Latin America. This wide impact is supported by the facts regarding the social involvement of partners. As examples, Universidad del Cauca has a Center for Open and Distance Learning offering courses in "Nasa" and "Guambiano" the two most important indigenous languages in the Paez and Guambian regions; Carlos III University has a special program for students with special needs that has finished its first edition and has been considered a complete success (just to mention some of these initiatives). These types of initiatives that have been proven effective are the ones considered in the framework of this project.

The consortium is also a well balanced set of institutions that include prestigious Latin American educational institutions (Universidad Galileo, Universidad de Chile, Universidad del Cauca, Universidad, Instituto Tecnológico y de Estudios Superiores de Monterrey), European partners with previous experience in successful projects (Trinity College of Dublin, University of Reading, Universidad Carlos III de Madrid, Institut National des Télécommunications), and the contribution of the research and development department of one of the most important telecommunications companies in Europe (Telefónica Investigación y Desarrollo).

These partners have previous experience in projects that required a level of interaction and collaboration similar to the one expected in E-LANE. Through the RICOTEL organization, Carlos III University has collaborated with numerous universities in Latin America, but more closely with the members of the consortium. Derived from this collaboration several initiatives have been taken that are similar in spirit to the ones in @LIS. Furthermore, the European partners have already a solid experience participating in successful projects within the same topic. University of Reading is the applicant and coordinator of the GENIUS project in which both Trinity College of Dublin and Carlos III University of Madrid are active partners. The objective of this on-going project funded by the European Union is to provide a detailed content for the

new ICT curricula proposed in the Career Space Consortium (www.career-space.com) and explore innovative instruction and delivery paradigms. Trinity College of Dublin has also participated in the European Community Driving License, an initiative to provide courses to facilitate the adoption of information technology by a wider portion of society. This initiative has been shown to be extremely successful, and therefore is an excellent model to be exported to Latin America. Furthermore, and as a continuation of the Genius project, a new proposal has been submitted with name "SEQUELS" that is oriented towards extending the new paradigms explored in the GENIUS context. It should be noted that all these three initiatives are in the context of e-learning, and therefore in line with this @LIS proposal.

Telefónica Investigación y Desarrollo, the industrial partner in the consortium participates in projects such as Atrium which uses the Geant network, planned to be extended in the near future to Latin American countries. Also, SIGECO, a commercial knowledge management system, has been partially funded by "Agencia de Desarrollo Económico de Castilla y León" (ADE). During 2001 and 2002, the amount of the funding from ADE was 90.000 euros per year, as part of annual programmes (www.eurada.org/Members/Spain/Castilla.htm).

University of Chile is involved in a project financed by a FONDEF grant supported by the Chilean government to use around 300 libraries in the country as an access point. The infrastructure for this project is financed by the Bill and Melinda Gates Foundation and the objective is to provide educational contents. The total budget is approximately 700.000US\$ over a 3 year period.

Derived from this intense interaction among the partners, a very fruitful trustworthy relationship has been built that allows for a very close and efficient interaction and coordination in the context of project development. The following figure illustrates how the proposed E-LANE project will benefit from the initiatives in which the partners are involved.

It is within the context of these projects that a wealth of interesting and effective techniques and methodologies has been developed that this project proposes to integrate and directly use in the context of e-learning in order to translate these advances to the Latin American countries. In fact, it is in these countries in which the impact can be maximized due to their intrinsic differences when it comes to access to education.

The needs in terms of education in Latin American countries should be addressed by taking into account the possibilities brought by technology. As a first classification, education could be divided into two categories, those courses oriented towards obtaining a basic diploma to contribute to bridge the **Digital Divide**, and those that are not oriented towards a diploma and contribute toward **Life Long Learning**. The former category provides basic concepts to be able to participate in the information society. This is the basis to help eliminate the "two gear" effect that is present in Latin America. Whereas certain countries have a high level of digital literacy, others have a very small fraction of the population in contact with information technology. This digital divide can be reduced by providing basic courses to bring technology to society as opposed to the other way around.

Life Long Learning emphasizes in a global and continuous perspective into education. As important to bringing the advantages of the information society to as many citizens as possible is extending these benefits to their entire life. An essential ingredient for a successful integration in the information society is the ability to learn concepts when required. This changes significantly the usual learning scenario with two phases, the first years in which certain concepts are acquired and the rest in which these concepts are applied.

In today's society this barrier is disappearing and both the learning and application phases are entirely overlapped. Life Long Learning is not simply a possibility but a necessity. But if the time a person spends learning new concepts and ideas increase, so should the efficiency of the process. In other words "the more we learn, the better we learn". Technology allows the learning process to be adapted to the student and therefore increase its efficiency. This process is usually based on some information obtained on how the student interacts with the course material. The desirable situation would be a learning platform capable of accessing the profile information of its users before they begin interacting with it. The course material could be adapted to better suit the needs of each student based on the information obtained from previous experiences.

It is for all these reasons that an integrated system for e-learning, a specially defined methodology and a set of courses is proposed as main objectives of the project.

As it can be seen in the rest of this document, the partners in Latin America all have the appropriate minimum infrastructure to undergo the tasks presented in this proposal. As an example, it follows the description of the level of connectivity present in each of the Latin American partners.

- Universidad Galileo: one E1 (2.048Mbps) connection to the Internet, which will be increased by an extra one in 2003 giving 4.096 Mbps. In Guatemala there are over 5 Internet Service Providers connected to the Internet at different bandwidths.
- Instituto Tecnológico y de Estudios Superiores de Monterrey: one E3 Internet connection configured in such a way that only 20 Mbps can be used, one E3 connection (34.368 Mbps) to Internet 2.
- Universidad del Cauca: two links to the Internet, one 1.5 Mbps and other of 1.0 Mbps for all university community. Soon an Internet2 (155 Mbps.) will be in operation.
- Universidad de Chile: one 10Mbps access to the Internet, and a 100Mbps access to the National IP network. Two big IP providers and many international operators (Impsat, Global Crossing, AT&T etc). The university of Chile administers the .CL top-level domain ([http:// www.nic.cl](http://www.nic.cl))

In Chile E-LANE will be aligned with a project already presented for a FONDEF grant (local Chilean government agency: CONICYT) to use the 300 libraries along the country as an access point [2]. The infrastructure is financed by the Bill and Melinda Gates foundation and our project is to provide the contents in education.

The ITESM in Mexico is well known for being a centre of excellence in e-learning with remote sites throughout Latin America. These will be used to further extend E-LANE courses, also after the EU funding.

There is a huge initiative on the part of the Ministry for Education in Brazil, aiming to distribute over 400,000 computers to primary and secondary schools nationwide, and have them connected to Internet. There is another initiative, sponsored by the International Development Bank (IDB), to organise a NGO-oriented branch of Internet in Brazil, including data bases, electronic conferences, etc., and empower NGOs of all types in the country.

In Guatemala, the government has set up a number of CIEs (Centros de Informática Educativa) throughout the country. The Universidad Galileo has also a number of courses of different levels ready for the platform which will serve as a basis for E-LANE.

9 Conclusions

The E-LANE project is well under way and has already produced a number of courses and tailored them to the open source platform. As the project continues through to implementation in the next twelve months many demonstrator courses will be delivered and most of these will be based on four modules of the ICDL (International Computer Driving Licence)

An agreement has been concluded with the ECDL Foundation, the governing body for the worldwide distribution of the ICDL which will see collaboration between the ECDL Foundation and the E-LANE Project team to ensure the take up of these courses throughout Latin America. The ICDL is already available in Argentina, Chile and Brazil. It will soon be available in Guatemala and Mexico. It is the intention to ensure availability in Colombia as soon as possible and to spread through the whole region in a short time period.

10. References

- [1] AOLserver, <http://www.aolserver.com>
- [2] BiblioRedes, <http://www.biblioredes.cl>
- [3] E-LANE, <http://www.e-lane.org>
- [4] European Computer Driving Licence, <http://www.ecdl.com>
- [5] IMS, <http://www.imsglobal.org>
- [6].LRN, <http://www.dotlrn.org>
- [7] OpenACS, <http://openacs.org>
- [8] PostgreSQL, <http://postgresql.org>
- [9] Creative Commons, <http://creativecommons.org/>
- [10] OSLO, <http://oslo.usu.edu/>

Edukalibre Project: Versatility in e-learning

Jaime Villate, Guilherme Dutra, Nuno Faria, Mário Lopes
Dep. de Física – Fac. de Eng. da Univ. do Porto
villate@fe.up.pt

Abstract

Edukalibre, a project funded by the Socrates/Minerva program of the European Union, aims at the promotion of information and communication technology in education. Its main goal is to transpose the advantages of the Free Software development model to the production of educational documentation.

This article focuses on the solutions found within the Edukalibre project to facilitate collaboration for the production of educational materials. The tools under development allow different authors to work on the same document, using different computer programs and therefore different document formats, without the need to rewrite existing content or to learn new formatting languages. The Edukalibre system also includes document version management and the possibility of writing documents on-line and/or off-line, via a user-friendly Web platform or through the Webdav protocol.

The results of a case study involving the elaboration of work reports by university students using the Edukalibre platform are also presented.

1. Introduction

The development of internet based communications has brought significant changes to the way software is developed. These are not merely technical changes. An entirely new concept of the information society has arisen. An example is the Free Software movement. More recently, these developments have led to increased interest in e-learning software tools such as Moodle (Moodle Project, 2004).

The Edukalibre project arises from these two concepts as it tries to introduce the free/libre software philosophy into the world of education document elaboration. Everyday, all around the world, different teachers teach the same course materials and write their own course manuals. All this effort could be better applied if these materials were created from cooperation between their authors. However, one of the main difficulties found is the wide range of typesetting tools available and their respective incompatible file formats. This article focuses on the solution found within the Edukalibre project to cope with these files.

2. The Edukalibre project

Edukalibre is a project financed by the Socrates / Minerva program (Edukalibre Project, 2005). Within this project, six European partners cooperate, trying to explore new ways to promote the elaboration of education materials in a way similar to the methodology generally used in communities devoted to the development of free/libre software.

One of the tasks of this project consists of developing software tools to facilitate the collaborative development of education-related documents like manuals and assignment reports. The idea is that a group of users should be able to work together, simultaneously on the same document, using the typesetting tool they prefer. The tool being developed is an interface to a collection of independent tools, namely:

- a document version control system;
- several document format conversion tools;
- several user interfaces among which are Moodle modules and OpenOffice macros.

Figure 1 illustrates the general architecture of the tool previously mentioned.

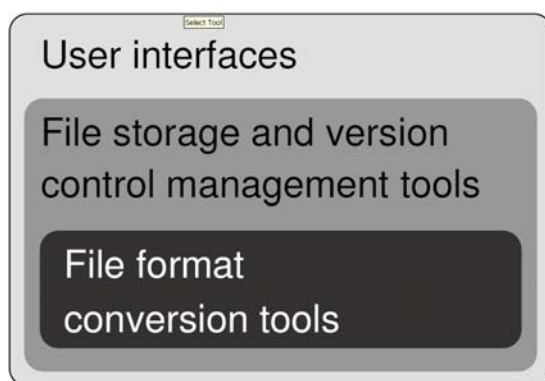


Figure 1. General architecture of the tools being developed under the Edukalibre project.

All software being used by the project is licensed as free/libre. Likewise, all tools developed by the project are made available to the community as free/libre software and all documents are published under a free/libre documentation license.

2.1. Document version management system

"Version control is the art of managing changes to information. It has long been a critical tool for programmers, who typically spend their time making small changes to software and then undoing those changes the next day. But the usefulness of version control software extends far beyond the bounds of the software development world", (Collins-Sussman et al., 2004).

In the Edukalibre system, version control is used to manage the documents. In fact, version control is the heart of the Edukalibre system. It deals with document storage, version management, creating and merging different version branches. The software chosen for this task was Subversion (Collins-Sussman et al., 2004). Some of the reasons that led to this choice were:

- Lock System: it forbids authors or collaborators to overwrite documents belonging other users;
- The documents may be cataloged using Name Spaces, thus it is possible to create collections of documents, even if they are in distinct places or have been erased.
- It is possible to add meta data, this information is stored in XML format transparently to the user.

2.1.1. Branches and hierarchy

The Edukalibre system assumes three levels of user hierarchies: Coordinator, Author, and Contributor/Reader. Figure 2 shows the hierarchical relation between people enrolled in a document editing.

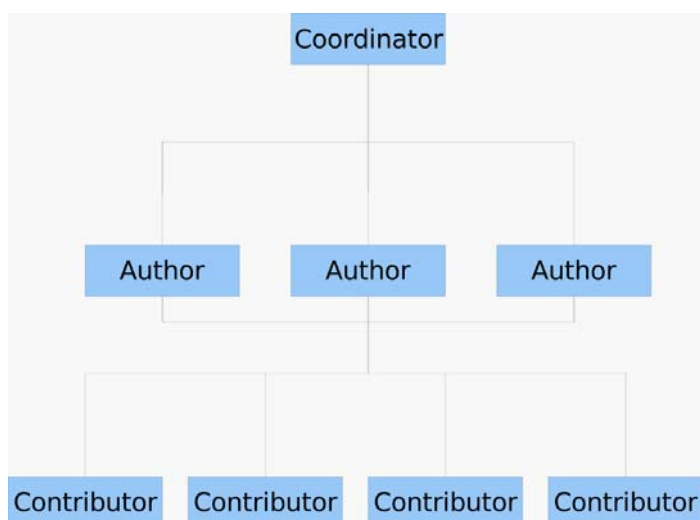


Figure 2. Hierarchy.

The Coordinator is responsible for managing the document edition. He decides who are the authors and when a new version of the document is released.

Authors can edit and create their own document versions. He decides whether or not to accept the contributions from Contributors. It can be a single author or a group of authors. Contributors/Readers are allowed to read, or edit and create their own document versions. These modifications start as new branches.

Each document has a main branch which is created and updated by a coordinator. When an author or contributor updates a document, a new branch is created. It is up to the coordinator to decide whether it will be merged or not into the main branch.

In figure 3 we can see an example that illustrates how the ramifications are treated. Some of them are merged in the main branch and others are not.

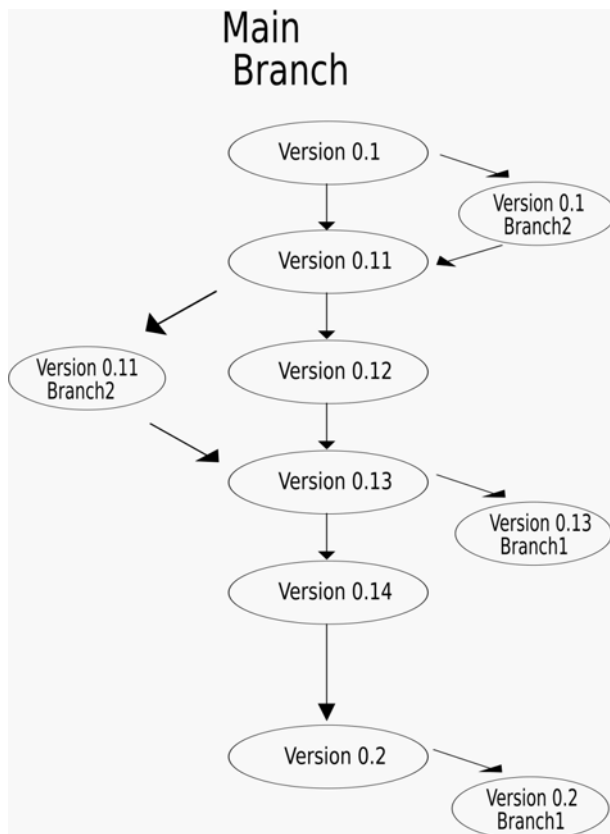


Figure 3. Branch schemas

2.2. Document typesetting tools and user interfaces

The users can access the repository through several alternative interfaces.:

- Command line, using an SVN client and SSL authentication;
- Direct access by Web-based Distributed Authoring and Versioning (WebDav), which is a set of platform independent HTTP extensions that allows users to edit and manage files in remote Web servers, (Webdav web site, n.d.);
- Transparent access through OpenOffice macros;
- Collab and Collab for Moodle - Web interfaces developed in PHP;
- Condor - A Moodle module which supplies a integrated set of tools useful for management of collaborative work.

Users can edit documents using OpenOffice, Wiki on-line editor, Lyx/Latex text editor, or even their preferred text editor, provided they change the XML source file itself. When a Wiki or OpenOffice document is uploaded to the repository, it is automatically converted to Docbook/XML format. Latex and other formats remain unchanged, see figure 4.

The Wiki on-line editor is available both on Collab and Condor Web interfaces. When documents are edited online, they are instantly posted into the repository. OpenOffice documents can be posted through OpenOffice macros, which are activated by menu, thus the post process is transparent for users.

Lyx/Latex and other formats must be posted by upload in Collab or Condor, or through an

SVN client.

Through the Collab interface it is possible to open and download all documents available in the repository. It is also possible to see the differences between the several versions of the documents. These capabilities are implemented using Webdav extensions.

The Condor module is integrated in the Moodle e-learning platform, (Moodle Project, 2005). This module includes, in the same interface, a message management tool, a real-time forum environment, and access to the Edukalibre repository.

3. Document format conversion tools

It was assumed from the beginning of the project that users should be allowed to choose the document typesetting tool they preferred. In order to do this, it was necessary to provide the system with a set of conversion tools for the formats allowed. This section describes the solutions found and the work still to be done. Figure 4 summarises the system architecture for document conversions.

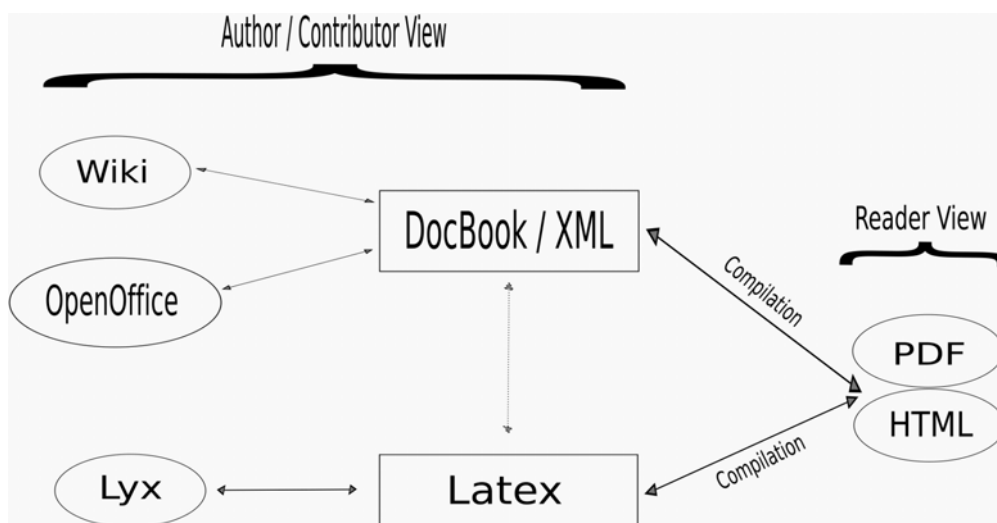


Figure 4. Conversions between formats.

3.1. Docbook XML - the base format

"Docbook is a very popular set of tags for describing books, articles, and other prose documents, particularly technical documentation. Docbook is defined using the native DTD syntax of SGML and XML", (Normal & Muellner, 2005).

Docbook is the Edukalibre system base format because it is open source, widely used for book and article writing, and it is possible, using OpenOffice, to write documents in Docbook format through a 'What You See Is What You Get' (WYSIWYG) interface. Moreover, there are conversions to several formats, and it is easy to write conversion scripts for XML (eXtensible Markup Language) documents.

From all the file formats allowed in the Edukalibre System, PDF, Postscript and HTML are only output formats. Everytime a new Docbook file is generated or uploaded, these formats are generated through the conversion tool FOP (The Apache XML Project, 2004).

3.2. Conversion to and from OpenOffice

OpenOffice.org (Sun Microsystems, 2002) is a free office productivity suite that differs from other tools in its way of handling and saving the documents data. Whereas some editors encode the content as a binary format, OO.org makes use of a XML-based format for storing it. This means that the information is always available in a human readable format.

One of the major advantages of this format is content, layout and meta information separation. The objects, such as images, are embedded into the final package. In Version 2 of the OO.org suite, the format has been adopted by OASIS (Organisation for the Advancement of Structured Information Standards) as a move to standardising it as a common format for saving office documents.

Although the file format is widely available, it is not yet being used widely. Fortunately, there are open and free software tools that convert to and from the OO.org document format.

In order to use them there are some tools available like OOoConv or BatchConv (Goddard, n.d.). Nevertheless, there are some other applications that manage to extend OO.org capabilities and convert to even more formats, like Writer2LaTeX (Just, n.d.) that is able to create a LaTeX file with BibTeX references.

As for the Edukalibre Conversion Tools framework, the default procedure is initially converting to Docbook by making use of a XSLT stylesheet and processor, and then making the other conversions from this (Figure 5).

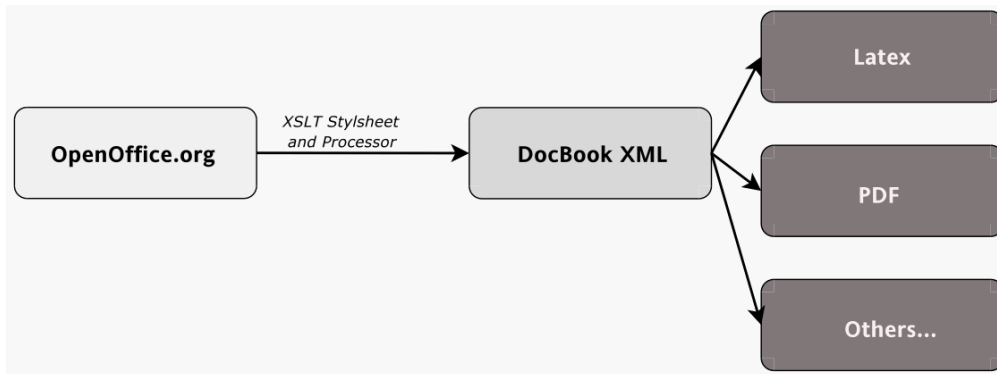


Figure 5. How Edukalibre handles OpenOffice.org

Unfortunately this cannot be done to OO.org files since Docbook is not as rich in formatting as OO.org. So, information and formatting styles would be lost in the conversion. Due to this, the conversion is made directly from OO.org using internal macros. This situation is represented in Figure 6.

The disadvantage of this method is that OpenOffice as well as a X Windows server (or a virtual frame buffer) must be running on the server. This will be addressed when OpenOffice 2.0 becomes a stable and more widespread platform.

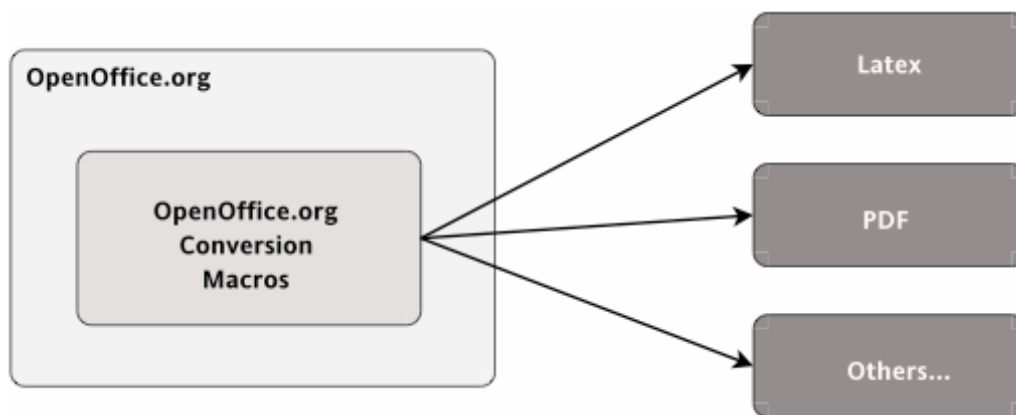


Figure 6. How Edukalibre handles OpenOffice.org conversions - detailed view.

3.3. Wiki - A format for online document typesetting

According to Wikipedia (Wikipedia, 2005), Wiki is an online collaboration model and tool that allows users to edit content through a simple browser. It is unnecessary to know tags, the syntax is comprised entirely of punctuation characters.

There is no single Wiki format standard. Instead, there are several similar formats but different enough to make it impossible to have one single fully compatible tool.

Within the Edukalibre project, the Wiki Markdown format (Gruber, 2004) was chosen. Its documents are easy to write and easy to read. The punctuation characters have been carefully chosen so that they convey, as much as possible, the formatting information they represent - asterisks around a word actually look like **emphasis**, blockquotes look like quoted passages of text and so on.

The TextWiki (Jones, 2004) package, which is a part of the PEAR (PHP Extension and Application Repository) framework (The PHP Group, 2004), has been used to convert Wiki to Docbook. It is an object oriented framework organised as *Parsers* and *Renderers*.

The *Parsers* analyse the text and identify Wiki rules occurrences.

When a Wiki rule is found, the Parser replaces the matched text with a "delimited token" and creates an entry in the tokens array. This process generates an intermediate document. The Renderers transform these tokens into its destination format equivalents, in this case, the Docbook format. Figure 7 illustrates this process.

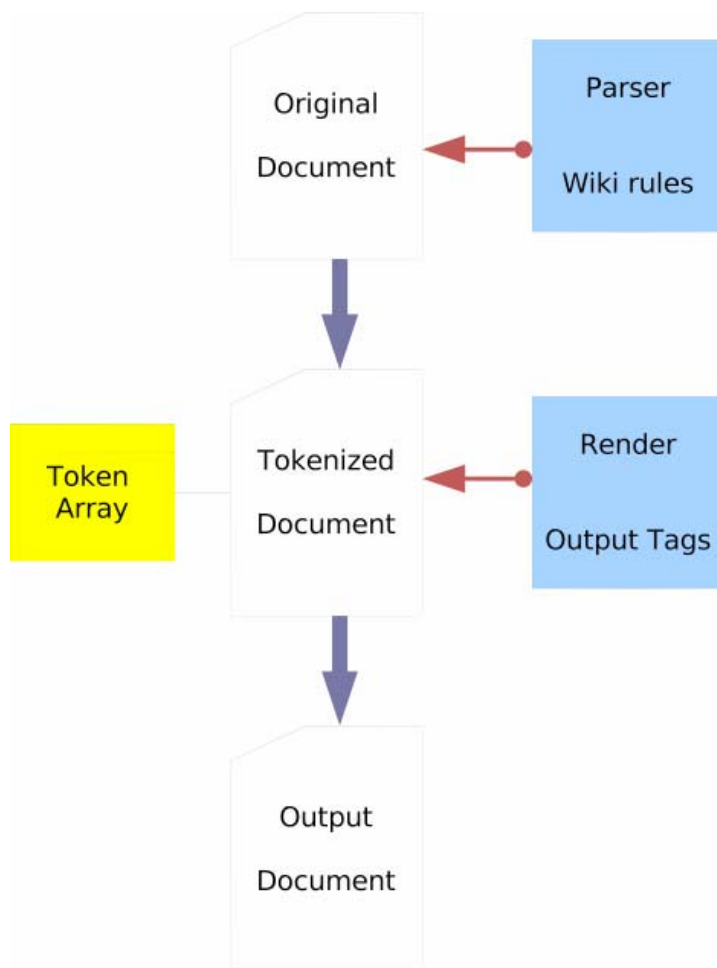


Figure 7. TextWiki schema.

In order to perform the Docbook to Wiki conversion, a new tool has been written from scratch. As the main language of the project is PHP, this tool was developed using the XML Parser and SimpleXML PHP toolkits.

As the Wiki format is simpler than the Docbook format, it is impossible to convert all Docbook text formatting tags. In order not to lose information from files which might have been generated with more complete tools, such as OpenOffice, it was decided to include these tags in the Wiki document, exactly as they appear in the Docbook original file.

3.3. Latex - A special case

Latex is an extremely rich typesetting language. Therefore, it is extremely difficult to develop a tool capable of converting any Latex source file to Docbook XML.

This reason could justify leaving the Latex format out of the Edukalibre project. However, Latex is widely used to produce scientific documents, especially those which involve writing complex mathematical equations.

Within the Edukalibre project, it was decided to include Latex as one of the allowed document formats. Until now only the conversion between Docbook XML and Latex is performed, using the "db2latex-xsl" stylesheet (db2latex Web Site, n.d.). If the original file is submitted to the system in Latex format, then only the output formats HTML, PDF and Postscript are generated. The Latex processor is in charge of producing the postscript and PDF files. The HTML version is generated by the program "latex2html" (latex2html web site, n.d.).

In the future the system may contemplate the conversion from Latex to Docbook, but this will certainly require the usage of a strict subset of Latex instructions.

4. Case Study

This case study focused on students from the Engineering Faculty of the University of the Oporto, who have little previous experience in collaborative writing (CW).

A group of four students from Informatics Engineering were asked to write a report about Artificial Intelligence. The content was split into three subjects. This task occurred from 2005-06-16 to 2005-06-22. The participants' average age was about twenty years.

4.1. Objectives

The aim of this study was to show that the adoption of a tool developed specifically to support collaborative writing stimulates and, consequently, improves the performance in the writing process. Moreover, it shows that the Wiki format may be used to write scientific documents.

4.2. Design

The collaborative writing process was supported by a Learning Content Management System (LCMS) web tool - Moodle. The document was written on-line and asynchronously through a Moodle module named WikiEduka, which consists of a Wiki text editor and an interface to the Edukalibre system.

As the Edukalibre system implements an SVN platform, it was possible to investigate which changes had been done by each user. The contributions of the participants were monitored by analysing the document versions. This analysis was done through the SVN Diff command, which shows what has been changed in each version.

The suggested writing strategy was Parallel Partitioning (Sharples et al., 1991), where each collaborator is responsible for a section. However, in this case, the others could always edit any section. The writing occurred in a distributed way, members could write anywhere (home, office, college laboratory, etc.). Since it was intended to be a hybrid experience, face-to-face meetings were also allowed.

The students had also been encouraged to make annotations to the document. These annotations gave rise to discussion of ideas, resolution of doubts, and decision-making through mutual dialogue. That communication took place via the chat and forum modules provided by Moodle.

4.3. Methodology

A questionnaire based on Noël and Robert (2004), composed of twenty questions aiming to gain the participants' previous experiences, had been previously presented. It confirmed that the participants were people with little experience in collaborative writing and showed what they expected from a collaborative writing tool. The details enclosed in the questionnaire had been: technologies previously used, subject complexity, work plan, leadership, communication, relationship, satisfaction, writing strategies, problems during project, ideas about ideal collaborative tool.

Productivity was evaluated through qualitative analysis of synchronous and asynchronous communication (chat and forum), and post quantitative/qualitative analysis of member contributions. The subjects measured were:

- Chat and Forum content (qualitative);
- Number of interventions (quantitative);
- Type of interventions (qualitative - quantitative);
- Interventions by subject;
- Document extension; total number of words (quantitative).

The type of intervention was determined through its summary. It was possible to distinguish among insertion, correction and cosmetic interventions. For changes without summary, when the number of new words was smaller than the average word number in insertion interventions, it was assumed that it was a correction intervention. Interventions with less than 4 words were considered cosmetic changes.

4.4. Analysis

The answers to the questionnaire showed that all participants had just little prior experiences with collaborative writing. In average, they had written two documents. None of them had used specific tools for collaborative writing, two of the students had used Microsoft Word, one had used Open Office and one had used Lyx (Latex). All of them pointed out that the communication and text grouping had been done via e-mail.

Analyzing the forum and chat contents we realized that there was little redundancy of ideas and that immediately after an item had reached consensus the document was modified.

Figure 8 shows that the number of words grew abruptly by the beginning of the period. This occurred because the first contributions were mainly new text insertions. After that, the number of words grew continuously during the rest of the testing period. It can be concluded that the participants developed a constant effort during the rest of the time.

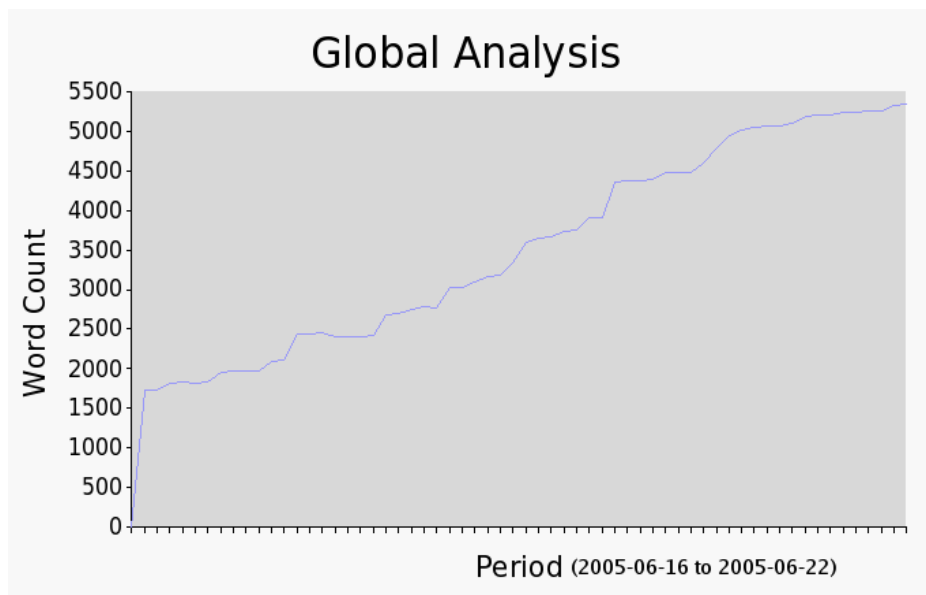


Figure 8. Global analysis

Figure 9 shows the evolution of the document by subject. Subject 1 had a beginning marked by new text insertions, but it was only improved after subject 2 had been sufficiently developed. New text insertions in subject 3 occurred at same time as subject 2 development, but corrections occurred at a moment of less activity in subject 2. The students focused their work on one subject at a time. This observation is probably explained by the fact that this was a reduced group.

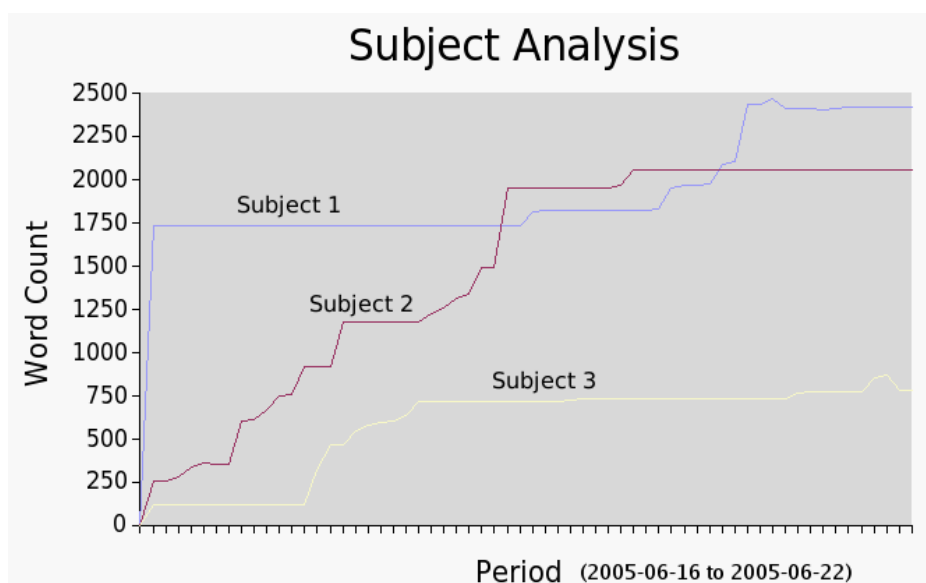


Figure 9. Subject analysis

Analyzing the nature of the interventions by each participant, it is possible to tell who preferred insertion of new text and who dedicated more time to corrections. Moreover it was possible to infer the amount of work dedicated by each participant (figure 10).

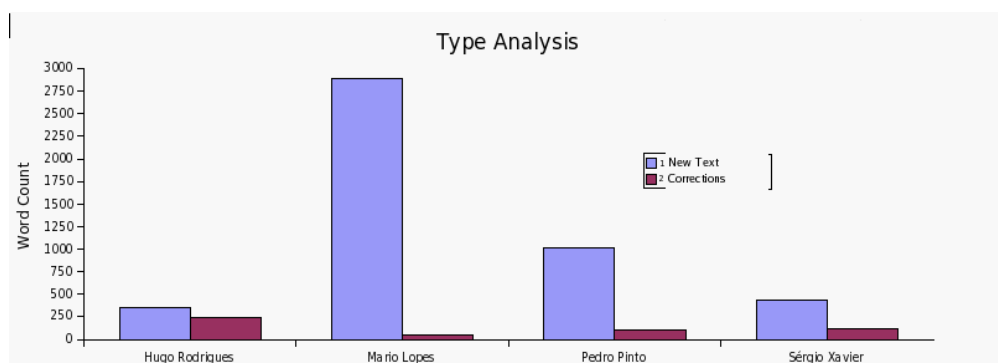


Figure 10. Type analysis

5. Conclusion

One of the most visible outputs coming out of the Edukalibre project is the Edukalibre System. This set of tools makes it possible for authors to collaborate over the Internet in the process of producing educational documentary materials.

Following the idea that led to the project in the first place, the choice of its underlying technologies took into consideration several criteria which came directly from the free/libre software philosophy. One of those guidelines consisted in using, as much as possible, tools previously developed and licensed as free software and also releasing all the tools developed within the project as free software. Moreover, in the spirit of the free software movement, it was decided from the beginning of the project that the authors should have the freedom to choose the typesetting tools they use. This led to the development of the file format conversion system described in this article.

The Docbook XML file format was chosen to be the main format from which all other formats are generated. This decision was taken, mainly due to the availability of several free software conversion tools, to and from this format. On the other hand, the Docbook format is oriented towards the document structure and lacks many of the formatting concepts allowed by more advanced typesetting tools such as OpenOffice and Latex. In order to overcome this set back, authors are encouraged not to use formatting information which is not included in the different templates developed for the Edukalibre system. Instead, the focus on the document structure should induce the production of standard formatted material which is better suited for wide dissemination.

One of the possibilities offered to authors is the on-line editing of the documents in a Wiki format. The Wiki Markup format has the advantage of being simple and fast to learn. However, it does not implement all the formatting options available in the Docbook XML format. In order not to lose formatting information when converting documents from Docbook to Wiki, it was decided to pass into the Wiki document all those Docbook formatting tags that have no equivalent in the Wiki format.

Future work related to document format conversions should concentrate on three main tasks. On one hand, finding an alternative way to convert from the OpenOffice format to Docbook without the need to execute it at the server is essential to promote the reliability of the server which will run the system. On the other hand, the conversion between Latex and Docbook is still a task to be undertaken. Finally, a more complete set of case studies should be carried out. These will not only allow an evaluation of the system but also the evolution of its goals and strategies.

References

- Collins-Sussman, B., Fitzpatrick W., B. & Pilato, C. M. (2004), "Version control with subversion", <http://svnbook.red-bean.com/en/1.0/svn-book.pdf>.
- db2latex Web Site (n.d.), <http://db2latex.sourceforge.net/>.
- Edukalibre Project (2004), "Edukalibre project web page", <http://www.edukalibre.org>.
- Godard, L. (n.d.), "OOoConv Web Page", <http://oooconv.free.fr>.
- Gruber, J. (2004), "Markdown", <http://daringfireball.net/projects/markdown/>.
- Jones, P. (2004), "Text wiki", [http://wiki.ciaweb.net/yawiki/index.php?area=Text Wiki](http://wiki.ciaweb.net/yawiki/index.php?area=Text+Wiki).
- Just, H. (n.d.), "Writer 2 LaTeX Web Page", <http://www.hj-gym.dk/~hj/writer2latex/>.
- latex2html web site (n.d.), <http://www.latex2html.org/>.
- Moodle Project (2004), "Moodle project web page", <http://moodle.org>.
- Noël, S. & Robert, J. (2004), "Empirical Study on Collaborative Writing: What Do Co-authors Do, Use, and Like?", *Comput. Supported Coop. Work* 13, 1, 63-89.
- Normal, W., Muellner, Leonard (2005), "Docbook 5.0: The definitive guide", <http://www.docbook.org/tdg5/en/html/docbook.html>
- Sharples, M., J. Goodlet, E. Beck, C. Wood, S. Easterbrook, L. Plowman & W. Evans (1991), "A Framework for the Study of Computer Supported Collaborative Writing", School of Cognitive and Computing Sciences, University of Sussex. Research Paper CSRP 190.
- Sun Microsystems (2002), "Open Office Org Web Page", <http://www.openoffice.org>.
- Tan, B. C. Y., Wei, K., & Lee-Partridge, J.E. (1999), "Effects of facilitation and leadership on meeting outcomes in a group support system environment", *Eur. J. Inform. Syst.*, vol. 8, no. 4, pp. 232-246.
- The Apache XML Project (2004), "Fop (formatting objects processor)", <http://xml.apache.org/fop/>.
- The PHP Group (2004), "Pear - php extension and application repository", <http://pear.php.net/>.
- Webdav web site (n.d.), <http://www.webdav.org/>.
- Wikipedia (2005), "Wikipedia, the free encyclopedia", <http://wikipedia.org>.

OMDoc: Open Mathematical Documents

Michael Kohlhase
 International University Bremen
 m.kohlhase@iu-bremen.de

Abstract

OMDoc (Open Mathematical Documents) was initially developed as a content-markup format for mathematical documents, where knowledge is marked up. On the:

Object Level it uses OpenMath and content MathML for objects represented as mathematical formulae;

Statement Level OMDoc provides original markup primitives that allow to specify the semantic structure and interdependencies of theorems, axioms, definitions, proofs;

Context-Level statements are grouped into mathematical theories, whose structure can be expressed by a rich set of theory morphisms. The semantic information embedded in OMDoc documents has for instance been used by eLearning systems to automate user-adaptation of course materials. Applying OMDoc to documents from science shows that only the object level has to be extended. The statement and context levels stay the same: they model the general "scientific method". Thus the extended three-level approach to knowledge representation can be used as an open basis for true eScience.

1. Introduction

A new generation of software systems is currently under development that provides integrated computer-based support for anybody who needs to find and wants to apply mathematical techniques – engineers, scientists, and also mathematicians. The applications range from the explanation of mathematical concepts and algorithms, over their search in large mathematical data bases, to E-Learning and applying the knowledge in the form of computation and reasoning. The distribution of information and services over the Internet has changed and will change all aspects of life, and science is not an exception. However, the systems making use of this information can only reach their full potential provided that they can interoperate to cover the whole work-flow of scientific research, education and application, which will work only if we manage to align the semantics of mathematical objects processed and communicated. Moreover, explicit semantics will allow providing important features which are indispensable for an effective usage of the search techniques, E-Learning systems, and applications which are currently in use.

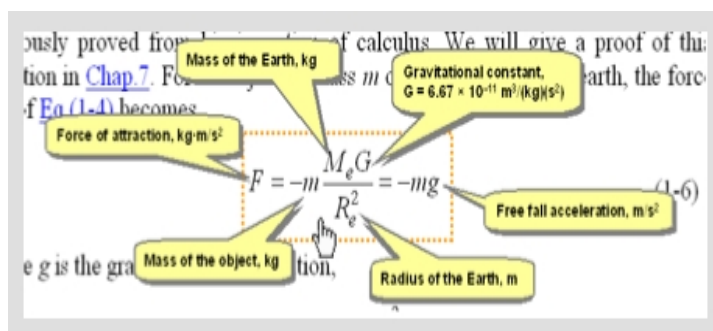


Figure 7: Mathematical formulas on the web

Most E-Learning environments are, nowadays, on the web — an indisputable quality for the ubiquity of the learning experience. Presenting mathematics on the web is however, still a challenge. On the one side, the MathML standard of the world-wide-web consortium, provides when browsers are supporting it an almost perfect solution for the layout of mathematics, on the other hand, value-added presentation of mathematical content is very limited within the bounds of this standard.

Value-added presentation includes interactive functionalities to enrich the display by indications of the formulae structures, the symbols' natures, or providing possibilities to further inspect the formulae presented.

It ranges up to the interactive presentation of formulae where formulae move along an explanation or where the formulae can be transferred for re-use in another part of the learner's environment. Most of these features can be provided by the semantic- or content-level encoding of the formulae and knowledge embedded in the texts. The structural properties of the encodings can also be used to personalize mathematical notation, for the composition of knowledge items into task-oriented- or didactically enhanced documents, and to fine-tune these by domain-independent user-modelling techniques [Melis et al., 2003].

Finally, semantic encoding of the mathematics also allows for the automated generation of multi-lingual features, describing the content in a language that is more understandable to students. Initial work in this direction is being pursued within the WebALT project (<http://www.webalt.net>) and aims at verbalizing mathematical content for several European languages.

In mathematics, the semantics of an object is determined by its structure (how is the object built up from already known objects, how is it defined in terms of other objects) and its context (what do we already know about these objects, how are these objects defined, what is their relation to other objects). Since the context information for mathematical objects (the mathematical knowledge about these objects) is dynamic and usually both large and highly structured, it would be very inefficient to communicate it along with the objects themselves. However, the context is crucial for an unambiguous understanding and can't be left to guessing. For this reason the context has to be represented and managed as an object in its own right. Typically a deep understanding of a particular area is necessary to understand the context correctly; and (as in everyday language, also in science) the task to understand a scientific concept correctly is difficult or impossible if the context is not clear. Since the context is given in the mathematical vernacular only seldom explicitly spelled out, its representation is a particular challenge.

In this paper, we present the OMDoc representation format (Open Mathematical Documents [Kohlhase, 2005]) for mathematical knowledge. This format is part of an initiative with the intention to develop, implement, and provide *semantic-based* and *context-aware* techniques for acquiring, organizing, processing, sharing and using knowledge in Sciences, Technology, Engineering and Mathematical disciplines (STEM) to support research, education, and technology application.

OMDoc follows and extends the approach of structural/semantic markup pioneered by the formula representation formats OPENMATH [Buswell et al., 2004] and MATHML [Ausbrooks et al., 2003]. Like these, OMDoc is now used in a large set of projects in automated theorem proving, eLearning, ePublishing, and in formal digital libraries. OMDoc extends the semantic representation format for mathematical formulae (OPENMATH objects or Content MATHML representations) by an infrastructure for context and domain models from "formal methods". In contrast to the latter, these structural/semantic approaches do not require the full formalization of mathematical knowledge, but only the explicit markup of important structural properties. For instance, a statement will already be considered as "true" if there is a proof object that has certain structural properties, not only if there is a formally verifiable proof for it. Since the structural properties are logic-independent, a commitment to a particular logical system can be avoided without losing the automatic knowledge management which is missing for semantically unannotated documents. Of course such representations support only structural plausibility checks for quality management instead of full verification. Work on the OMDoc format shows that most added-value services in knowledge management do not need tedious formalization, but can be based on the structural/semantic level. OMDoc does not take the all-or-nothing approach of the traditional theorem proving community that either guarantees full correctness of a theorem, or does not give any support. The OMDoc format rather provides added value, which supports users on different levels of precision.

With the OMDoc format we are proposing a three-layered structure model for semantic representation formalisms:

Object level: represents objects such as complex numbers, derivatives, etc. for mathematics, map specifiers for geo-sciences or observables for physics. Semantic representation formats typically use functional characterizations that represent objects in terms of their logical structure, rather than specifying their presentation. This avoids ambiguities which would otherwise arise from domain specific representations.

Statement Level: (natural/social/technological) sciences are concerned with modeling our environment, more precisely with statements about the objects in it. We can distinguish different types of statements: model assumptions, their consequences, hypotheses, and measurement results. All of them have in common that they state relationships between scientific objects and have to be verified or falsified in theories or experiments. Moreover, all these statements have a conventionalized structure, such as Exercise, Definition, Theorem, Proof, and a standardized set of relations among each other. For instance, a model is fully determined by its assumptions (also called *axioms*); all consequences are deductively derived from them (via *theorems* and *proofs*), and therefore their experimental falsification uncover false assumptions of the model.

Theory/Context Level: Representations always depend on the ontological context; even the meaning of a single symbol⁵¹ is determined by its context, and depending on the current assumptions, a statement can be true or false. Therefore the sciences (with mathematics leading the way) have formed the habit to fix and describe the situation of a statement. Unfortunately, the structure of these situation descriptions remain totally implicit, and can therefore not be used for computer-supported management. Semantic representation formats make this structure explicit. For instance in mathematical logic, a theory is the deductive closure of a set of axioms, i.e. the (in general infinite) set of logical consequences of the model assumptions. Even though this fully explains the phenomenon context in theory, important aspects like the re-use of theories, knowledge inheritance, and the management of theory changes are disregarded completely. Therefore, formalisms with context level use elaborate inheritance structures for theories, e.g. in form of ontologies in the Semantic Web or in form of "algebraic specifications" in program verification.

An important trait of the three-layer language architecture is the inherent dependency loop between the object- and theory levels mediated by the statement level: the objects obtain their meaning from the theories their functional components are at home in, and the theories are constituted by special statements, and in particular the objects that are contained in them. This structure implicitly pervades the scientific discourse (hence the name "scientific method") and the whole corpus of scientific knowledge. To make these structures explicit enables the mechanization and automation of knowledge management and the unambiguous, flexible communication of mathematical objects and knowledge that is needed for meaningful interoperability of software systems in science.

51. e.g. the glyph h as the height of a triangle or Planck's quantum of action.

2. The OMDoc format

To achieve content and context markup for mathematical knowledge, OMDoc uses three levels of modeling corresponding to the concerns raised previously. We have visualized this architecture in

Figure 8.

Level	Example
Object level: OPENMATH/Content-MATHML <ul style="list-style-type: none"> Objects as logical formulae Semantics by pointing to theory level 	<pre><OMA> <OMS cd="arith1" name="plus"/> <OMV name="X"/> <OMS cd="nat" name="zero"/> </OMA></pre>
Statement level: <ul style="list-style-type: none"> Definition, Theorem, Proof, Example Structure explicit in forms and references 	<pre><definition for="#plus" type="rec."> <CMP>rec. eq. for plus</CMP> <FMP>X+0=0</FMP> <FMP>X+s(Y)=s(X+Y)</FMP> </definition></pre>
Theory level: Development Graph <ul style="list-style-type: none"> Inheritance via symbol-mapping Theory inclusion by proof-obligations Local (one-step) vs. global links 	<p>The diagram illustrates the Theory level Development Graph. It shows four theory nodes: Nat-List (top-left), List (top-right), Nat (bottom-left), and Param (bottom-right). Each node contains a list of symbols: Nat-List (cons, nil, 0, s, Nat, <), List (cons, nil, Elem, <), Nat (0, s, Nat, <), and Param (Elem, <). Relationships are shown as follows: Nat-List imports List (solid blue arrow); List imports Param (solid blue arrow); Nat imports Param (solid blue arrow); Nat-List imports Nat (solid blue arrow); List imports Nat (dashed blue arrow); Nat-List actualizes List (solid red arrow); Nat-List has proof obligations to List (yellow oval labeled 'Proof Obligations' with a dashed red arrow); List has proof obligations to Param (yellow oval labeled 'Proof Obligations' with a dashed red arrow); Nat-List has theory-inclusion to List (dashed red arrow); List has theory-inclusion to Param (dashed red arrow).</p>

Figure 8: OMDoc in a Nutshell (the three levels of modeling)

Object Level At the lowest level of mathematical formulae, OMDoc uses the established standards OPENMATH [Buswell et al., 2004] and Content-MATHML [Ausbrooks et al., 2003]. These provide content markup for the structure of mathematical formulae and context markup in the form of URI references in the symbol representations.

Statement Level OMDoc provides an original markup scheme for making the structure of mathematical statements explicit. Again, we have content and context markup aspects. For instance the definition in the second row of

Figure 8 contains an informal description of the definition as a first child and a formal description in the two recursive equations in the second and third children supported by the type, which states that this is a recursive definition. The context markup in this example is simple: it states that this piece of markup pertains to a symbol declaration for the symbol plus in the current theory (presumably the theory arith1).

Context Level At this level, OMDoc supplies original markup for clustering sets of statements into theories, and for specifying relations between theories by morphisms. By using this scheme, mathematical knowledge can be structured into reusable chunks. Theories also serve as the primary notion of context in OMDoc, they are the natural target for the context aspect of formula and statement markup.

All levels are augmented by markup for various auxiliary information that is present in mathematical documents, e.g. notation declarations, exercises, experimental data, program code, etc.

3. Situating the OMDoc Format

The space of representation languages for mathematical knowledge reaches from the input languages of computer algebra systems (CAS) to presentation markup languages for mathematical vernacular like $T_E X/L^A T_E X$.

We have organized some of the paradigmatic examples in a diagram mapping coverage (which kinds of mathematical knowledge can be expressed) against machine support (which services the respective the software system can offer).

Computer algebra systems like MATHEMATICA or MAPLE are relatively restricted in the mathematical objects — they can deal with polynomials, group representations, differential equations only, but in this domain they can offer sophisticated services like equation solving, factorization, etc. More to the right we see systems like automated theorem provers, whose language — usually first-order logic — covers much more of mathematics, but that cannot perform computational services⁵² like finding solutions like the CAS.

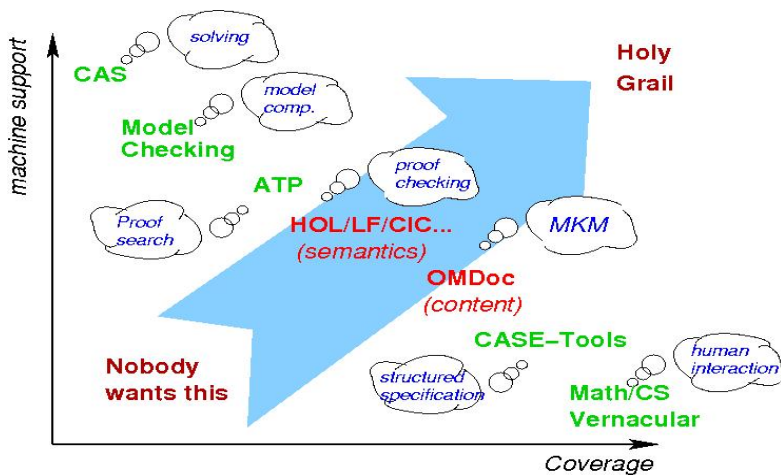


Figure 9: Situating Content Markup: Mathematical Knowledge Management

In the lower right hand corner, we find languages like the “mathematical vernacular”, which is just the everyday mathematical language. Here coverage is essentially universal: we can use this language to write international treaties, math books, and love letters; but machine support is minimal, except for typesetting systems for mathematical formulae like T_EX, or keyword search in the natural language part.

The distribution of the systems clusters around the second diagonal. This already suggests that there is a trade-off between coverage and mechanization. All of the representation languages occupy legitimate places in the space of representation languages, trying to find sweet-spots along coverage/support trade-off. OMDoc tries to occupy the “content markup” position. To understand this position better, let us contrast it to the “semantic markup” position immediately to the left of and above it. This is an important distinction, since it marks the border between formal and informal mathematics.

We define a **semantic markup format** (also known as: ‘formal system’) as a representation system that has a semantics, i.e. a system that has a way of specifying when a formula is a consequence of another. Many semantic markup formats express the consequence relation by means of a formal calculus, which allows the mechanization of proof checking or proof verification. It is a widely held belief in mathematics, that all mathematical knowledge can in principle be expressed in a formal system, and various systems have been proposed and applied to specific areas of mathematics. The advantage of having a well-defined consequence relation (and proof-checking), has to be paid for by committing to a particular logical system.

Content markup does not commit to a particular consequence relation, and concentrates on providing services based on the marked up structure of the content and the context. Consider for instance the formulae in

Figure 8, where the OPENMATH representation does not specify the full consequence relation (or the formal system) for the formula. It does something less but still useful, which is

52. Of course, in principle, the systems could, since computation and theorem proving are irreducible, but in practice theorem provers get lost in the search spaces induced by computational tasks.

what we could call *semantics by pointing*: The symbols used in the representation are identified by a pointer (the URI jointly specified in the `cd` and `name` attributes) to a defining document (in this case an OPENMATH content dictionary). Note that URI equality is a sufficient condition for two symbols to be equal, but not a necessary condition: Two symbols can be semantically equal without pointing to the same document, e.g. if the two defining documents are semantically marked up and the definitions are semantic consequences of each other.

In this sense, content markup offers a more generic markup service (for all formal systems; we do not have commit ourselves) at the cost of being less precise (we for instance miss out on some symbol equalities). Thus, content markup is placed to the lower right of semantic markup in

Figure 9. Note however, that content markup can easily be turned into semantic markup by adding a consequence relation, e.g. by pointing to defining documents that are marked up semantically. Unlike OPENMATH and Content-MATHML, the OMDoc format straddles the content/semantics border by closing the loop and providing a content markup format for both formulae and the defining documents. In particular, *an OMDoc document is semantic if all the documents it references are*.

As a consequence, OMDoc can serve as a migration format from formal to informal mathematics. A document collection can be marked for content and context structure, making the structures and context references explicit in a first pass. Note that this pass may involve creating additional documents or identifying existing documents that serve as targets for the context references so that the document collection is self-contained. In a second (and possible semi-automatic) step, we can turn this self-contained document collection into a formal representation (semantic markup) by committing on consequence relations and adding the necessary detail to the referenced documents.

4. An active Web of Mathematical Knowledge

It is a crucial – if relatively obvious – insight that true cooperation of mathematical services is only feasible if they have access to a joint corpus of mathematical knowledge. Moreover, having such a corpus would allow developing added-value services like

1. Cut and paste on the level of computation (take the output from a web search engine and paste it into a computer algebra system),
2. Automatically proof checking published proofs,
3. Math explanation (e.g. specializing a proof to an example that simplifies the proof in this special case),
4. Semantical search for mathematical concepts (rather than keywords),
5. Data mining for representation theorems (are there unnoticed groups out there?),
6. Classification: Given a concrete mathematical structure, is there a general theory for it?

As the online mathematical knowledge is presently only *machine-readable*, but not *machine-understandable*, all of these services can currently only be performed by humans, limiting the accessibility and thus the potential value of the information. Services like this will transform the now passive and human-centred fragment of the Internet that deals with mathematical content, into an active (by the services) web of mathematical knowledge. Of course, this promise of activating a web of knowledge is in no way limited to mathematics, and the task of transforming the current presentation-oriented world-wide web into a “semantic web” [Lee, 1998] has been identified as one of the main challenges by the world wide web consortium (W3C).

5. Conclusion

Mathematics is a fundamental discipline in science, technology, engineering, and society. Fundamental problem solving skills, acquired when learning mathematics properly, such as structuring a problem or a situation, analyzing pre-requisites and implications, go beyond doing mathematics and influence social capabilities and behaviour. Yet, the number of freshmen in mathematics has been decreasing all over Europe in the past 10 years and as a

result, there is a lack of new mathematicians able to entice new generations in primary school and to tutor university students.

One way to overcome the crisis of mathematics education is to improve the didactic tools and the lecture material. Computational software can be readily accessed by the students and used to practice a variety of skills, ranging from manipulation and evaluation of formal expressions, to proving that a theorem holds in a certain theory, to plotting colourful surfaces in order to highlight specific characteristics in a physics experiment. Integrating, in a teaching environment, the variety of computational software packages that handle some form of mathematical data is possible only if the underlying mathematical descriptions are given in an unambiguous semantic encoding.

The OMDoc representation format provides an open content markup scheme for mathematical knowledge and documents and has attracted several e-Learning systems that use it as a representational basis for providing semantic search [Franke and Kohlhase, 2000, Kohlhase and Franke, 2001], personalized course materials and user modelling [Melis et al., 2003], formal software engineering [Hutter et al., 2005], and distributed automated theorem proving [Zimmer, 2004].

References

- [Ausbrooks et al., 2003] Ausbrooks, R., Buswell, S., Carlisle, D., Dalmás, S., Devitt, S., Diaz, A., Froumentin, M., Hunter, R., Ion, P., Kohlhase, M., Miner, R., Poppelier, N., Smith, B., Soiffer, N., Sutor, R., and Watt, S. (2003). Mathematical Markup Language (MathML) version 2.0 (second edition). W3c recommendation, World Wide Web Consortium. Available at <http://www.w3.org/TR/MathML2>, 2005.
- [Buswell et al., 2004] Buswell, S., Caprotti, O., Carlisle, D. P., Dewar, M. C., Gaetano, M., and Kohlhase, M. (2004). *The Open Math standard, version 2.0. Technical report*, Available at: The Open Math Society. (<http://www.openmath.org/standard/om20>).
- [Franke and Kohlhase, 2000] Franke, A. and Kohlhase, M. (2000). System description: MBASE, an open mathematical knowledge base. In: McAllester, D., (ed), *Automated Deduction – CADE-17*, number 1831 in LNAI, p. 455–459. Springer Verlag.
- [Hutter et al., 2005] Hutter, D., Mossakowski, T., and Schairer, A. (2005). MAYA: Maintaining structured developments. To appear in [Kohlhase, 2005].
- [Kohlhase, 2005] Kohlhase, M. (2005). OMDoc an open markup format for mathematical documents (version 1.2). Manuscript, <http://www.mathweb.org/omdoc/omdoc1.2.ps>. To appear in Springer LNAI.
- [Kohlhase and Franke, 2001] Kohlhase, M. and Franke, A. (2001). MBase: Representing knowledge and context for the integration of mathematical software systems. In: *Journal of Symbolic Computation; Special Issue on the Integration of Computer algebra and Deduction Systems*, 32 (4), p. 365–402.
- [Lee, 1998] Lee, T. B. (1998). The semantic web. W3C Architecture Note. Available at: <http://www.w3.org/DesignIssues/Semantic.html>, 2005.
- [Melis et al., 2003] Melis, E., Büdenbender, J., Gogvadze, G., Libbrecht, P., and Ullrich, C. (2003). Knowledge representation and management in activemath. In: *Annals of Mathematics and Artificial Intelligence*, 38, p. 47–64. See also: <http://www.activemath.org>
- [Zimmer, 2004] Zimmer, J. (2004). A Framework for Agent-based Brokering of Reasoning Services. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., and Azuela, J.H.S., (eds), *MICAI*, volume 2972 of *Lecture Notes in Computer Science*. Springer.

Open Content and Source: European Schoolnet Riding the Wave

Riina Vuorikari, Karl Sarnow
European Schoolnet
Riina.Vuorikari@eun.org, Karl.Sarnow@eun.org

Abstract

This paper summarises the actions that European Schoolnet and its members have taken in the area of open content and free and open source (FOSS) development in the field of e-learning. European Schoolnet provides insight into the educational use of information and communications technology (ICT) in Europe for policy-makers and education professionals. This goal is achieved through communication and information exchange at all levels of school education using innovative technologies, and by acting as a gateway to national and regional school networks. In the recent years European Schoolnet and some of its member networks have, little by little, started trends towards awareness building, piloting and the rolling-out of open source software programmes for schools, as well as investigating possibilities in the area of open content. Advances have been made in all the areas from analysing and visibility studies to the development of educational software based on open source.

1. Introduction

European Schoolnet is a network of 26 European Ministries of Education, founded in 1996 with the mandate of the Council of the European Union. European Schoolnet (EUN) promotes the use of information and communication technologies (ICT) in European schools, acting as a gateway to national and regional school networks. The mission is two-fold: on the one hand EUN works closely with national and regional policy-makers and shapers by setting up special interest committees, involving them in transfer of best practices, and in research and development. On the other hand, EUN works directly with a large network of European schools through special on-line events organised in collaboration with a variety of stakeholders.

European Schoolnet is committed to following open standards in e-learning research and development that it conducts in the field, partnering up with different stakeholders from public, private and industry partners. This has resulted in services that allow multiple players access to the field. Furthermore, the use and development of open source software in education is becoming more of a concern, whereas the promotion of interoperable content-based services such as federations of learning resource repositories has long been in the centre of EUN's attention.

This paper introduces actions that European Schoolnet and its members have taken in the area of open content, and free and open source software (FOSS) development in the field of ICT and education. The paper has four main focuses. First, it will describe European Schoolnet's recent development in the field of content provision, focusing in promoting a rightful use and re-use of educational content. This section explains the implementation of Digital Rights Management Framework in EUN's current educational content network development. Also, it explores the current development of a Learning Toolbox to support collaborative learning based on open source.

The second main section of the paper looks at some emerging campaigns led by a number of EUN's members. This section describes the promotion of the use of open source software in Belgium's Flemish Community, in the Netherlands, in Estonia, in the UK, Slovenia and Ireland. The third section presents the Xplora project that promotes science education in European schools. Xplora takes a stance towards the true nature of science - sharing open source educational software for science. Finally, the paper draws an outline of EUN's Special Reports service where policy briefs can be found on the issues dealing with open source and content in education.

2. EUN content services in pipeline

Since 2000, European Schoolnet has led EU funded projects to provide better access to digital educational content for teachers and learners across Europe. The CELEBRATE project (2002-2004) provided the first large-scale demonstration and evaluation of Learning Object (LO) interoperability and the use of LOs in schools at a European level.

In 2004, a survey of thirteen Ministries of Education participating in European Schoolnet also indicated that they wished to take forward the vision of a European Learning Resource Exchange (LRE) based on the architecture demonstrated in the project. Furthermore, many communicated that LOs are increasingly seen as an important, and in some cases a key component, in the content development strategies of Ministries of Education. Also, the majority expressed interest in open source content development strategies where "Learning Object economies" were created for open source and commercial content to co-exist.

EUN continues its work towards an enhanced architecture for learning resources in Europe. Within a recent 6th Framework Programme, the CALIBRATE project was funded. EUN will continue to lead the development of the LRE based on a Brokerage System Architecture (of which the code for the Brokerage System is licensed under the LGPL) involving a variety of stakeholders from content providers, both public and commercial, to end-users in European schools.

A set of more tailored services will be offered to the members of LRE such as federated searches, learning resource exchanges, and digital rights management. It is envisaged to support multiple digital right expression languages and permit content providers to select the level of digital rights management that best fits their needs in terms of intellectual property protection. This requires a proper Digital Rights Management (DRM). The objective is to design and implement a DRM framework that takes into account requirements from all stakeholders, thus supporting available DRM standards like ODRL and Creative Commons.

European Schoolnet supports the use of Creative Commons licenses within its services and have already implemented an integrated interface for its users to choose an option of Creative Commons license for the resources that they submit to various EUN projects.

2.1. Open source learning toolbox to support collaborative learning

European Schoolnet's research into the use of learning environments confirms that a number of its members favour the development of open source VLEs. Moreover, many expects the next generation of new learning platforms to facilitate the adoption of more learner-centred and collaborative pedagogical approaches. However, the same survey and subsequent observations suggest that these high expectations are not yet being met. Most teachers are still using VLEs as little more than a "digital distribution" space, somewhere to upload store and distribute content and to issue assignments to students.

Within the above mentioned CALIBRATE project, EUN will lead the development of a VLE which brings together two quite distinct and somewhat opposing methodologies for pedagogical affordance; the first comes from a background of social constructivist pedagogies and collaborative knowledge building, whereas the second has a background in SCORM and LCMs. By drawing on both these approaches, a new open source toolbox will be built using the existing code from the Future Learning Environment 3 (FLE3) based on Plone/Zope. The VLE will offer a richer feature set and which will be developed with the help of practicing teachers.

3. Start of the tidal wave: EUN's members actions in the quest for educational open source software

A number of European Schoolnet's partners have explicit roles in promoting the use and development of open source software as an alternative choice for schools. A review of a selection of partners acting upon this challenge is provided in this section presenting the Ministry of the Flemish Community, the Education department in Belgium, Kennisnet in the Netherlands, Becta in the UK, Tiger Leap foundation in Estonia, the Ministry of Education and Sport in Slovenia and the National Centre for Technology in Education in Ireland.

3.1. Belgium: The Ministry of the Flemish Community, Education department

The Ministry of the Flemish Community, Education department in Belgium, has an explicit role in promoting the use of open source software as an alternative choice for schools. In 2004 the former Minister of Education, Ms Marleen Vanderpoorten, commissioned an advice on the issue, which led to a vision and a proposed action plan.

In 2005, a large campaign has been organised to introduce free and open source software (FOSS) in Flemish schools aiming to highlight its educational possibilities. In this campaign, a publication, a CD and an educational tools database are drafted and a conference is organised.

By means of the publication, "Free software in education", a practical guide for the use of FOSS and open educational tools is spread amongst all schools. Beside general information on the "what and how" of FOSS, one finds descriptions of a number of interesting open source applications. In association with the educational portal Klascement, an educational tools database was developed for these applications. This is also the general campaign website. Additionally, a conference was organised addressing FOSS and open educational tools targeting audiences such as teachers, headmasters and ICT co-ordinators.

Furthermore, the Education Department in Flanders has created didactic sheets on the use of educational freeware and open content based on the primary education curriculum topics. The didactic sheets have been published as a book, "ICT on the menu", and are searchable in a database through the portal. The scenarios are a helpful means to make the ICT integration in primary education more concrete. In 2005, a similar project was developed for secondary education. This time, the work was carried out by teachers from the secondary ENIS schools. The result is a publication, both on paper and on-line, called "Digital resources for secondary education. In 2006 a CD will be published with open learning tools and open source educational software that is currently under a validation process by the European Network of Innovative Schools (ENIS).

3.2. The Netherlands: Open source and open standards in education (OSS in het onderwijs)

Since 2003, in the Netherlands the government has brought open standards and open source into the central focus of its attention. A variety of initiatives have been set up to work on cross-sectoral issues that touch upon open standards as well as open source development. As for the education, there are initiatives, programmes and actions taken to foster the efforts in the field and to muster the common efforts.

The programme 'OSS in het onderwijs', translated as open source and open standards in education, is a joint initiative between Kennisnet, ICT op School and a government wide programme called OSSOS, the Programme for Open Standards and Open Source Software. Additionally, to involve a diversity of partners in the field of education, a new association called EduStandaard has been set up. The association aims to manage the standards that are used in the Dutch educational field, comprising stakeholders such as publishers, schools etc. Also, Kennisnet promotes a programme to improve the use of open standards for content.

A central point for 'OSS in het onderwijs' is a webspace where the Dutch education community can discuss open source and open standards, aiming at both novices and experts. The main focus is on primary and secondary education, but also on the field of vocational training. The programme is informative, aiming at offering alternative solutions for schools who have an independent budget to spend on educational technologies. The programme targets mainly the IT coordinators, administrators, and teachers who are responsible for IT set-ups in schools, but also at teachers who use computers and ICTs in their lessons.

One powerful means to transfer good practices and ideas of the use of FOSS in education is the case studies that can be found at the website of 'OSS in het onderwijs'. These case studies are simple descriptive interviews with practitioners on topics such as how to use GIMP for manipulating images, etc.

Moreover, 'OSS in het onderwijs' has prepared an information package on a CD ROM that focuses on the use of open standards in all processes in school that can involve the use of information technologies, from administrative tasks to using applications for learning purposes, gathering content about the student for portfolios, as well as other actions for the creation, exchange and altering of the content. The idea was to identify all actions and propose alternatives where closed systems or standards are used. This aims at better overall interoperability within schools' information systems. The CD will be released in November 2005. In the same spirit, a booklet on open source software was created in 2004 for schools. These information packages can be requested from the website, but they are also handed out at local ICT conferences.

Furthermore, the programme 'OSS in het onderwijs' can help schools to implement open source, not only in advisory terms, but they can make small amounts of money available to pay for a third party programmer or consultant to, for example, find compatibility solutions between an existing system and the new one based on open source and standards. On the website, there is also a FOSS helpline for schools to help them to solve small scale problems. In this regard, the programme tries to match the need that schools have for support with existing supply in the market. On the website one can find an overview of companies with experience of FOSS and education.

'OSS in het onderwijs' has been running for three years, 2005 being the final year with a big push. The continuation for the next year is still unsecured. A conference with one day education track on the topics will be held on December 8 2005.

3.3. The UK: Evaluation of open source software in schools

In May 2005, the British Educational Communications and Technology Agency (Becta), released an evaluation of the use of open source software within a number of schools. In the UK, some previous government studies have suggested that the use of OSS within the UK public sector can provide a viable and credible alternative to propriety software and lead to significant cost savings.

The study, funded by the Department for Education and Skills, had three main aims: to examine how well the open source software approach works, compared with proprietary offerings, in supporting delivery of the school curriculum and administration; to compare the total cost of ownership (TCO) of using OSS within school environments against that of non-open-source solutions; and to highlight examples of successful school-based open source implementations.

The report, '*Open Source Software in Schools: A study of the spectrum of use and related ICT infrastructure costs*', demonstrates that although the implementation of OSS in schools needs careful planning and support, it can offer a cost-effective alternative to proprietary software. For the way forward with FOSS, the report examines cost-effective models of support for OSS in schools, best practice in licensing solutions, successful implementation to run the school's servers, to provide school-wide facilities, operating systems and administrative PCs, and OSS applications on classroom and administrative PCs.

According to Becta's Chief Executive, Owen Lynch, Becta believes that software used in schools should be of a high quality and adhere to open standards, enabling compatibility and interoperability between products. Becta will now be undertaking more extensive research across a wider range of institutions to allow further analysis of these issues.

3.4. Estonia: the Tiger Leap Foundation (Tiigrihüppe Sihatusutus) financing educational FOSS development

In 2004, in Estonia, the Tiger Leap Foundation (TLF) initiated a project for distribution and promotion of freeware in schools, aiming to be launched at county level in the autumn of 2004. In the course of the project, it is foreseen to release a Linux distribution that is suitable for schools, to prepare training materials and to train teachers. Furthermore, since the spring 2005, TLF only supports projects which will be released under General Public License for the code and for the content a Creative Commons License will be required.

A number of Estonian educational open source software applications have been developed with the financial support of TLF in collaboration with Tallinn University (TU). The development of the Virtual learning environment VIKO started in 2001. Schools do not have

to set up their own server, VIKO is offered as a free service by Tallinn University. Furthermore, KooliPlone, a Plone-based Content Management System for school websites is also developed in TU, the expected release is in autumn 2005.

Another large scale development of a Learning Management System called IVA was supported by Estonian Ministry of Education and Science, the Estonian Information Technology Foundation and Hansapank, the largest bank in Estonia. IVA is also developed in TU, based on Zope and an existing educational platform called Fle3. It has Estonian, Russian and English user interfaces and is currently used by more than 2000 users in TU.

Additionally, Estonia being a country representing a small market, the government has funded the translation of OpenOffice's spell-check programme in Estonian.

3.5. Slovenia: actions by the Ministry of Education and Sport

The Slovenian Ministry of Education and Sport has a focus on three main areas providing basic tools, didactic tools and promoting open source for teachers, headmasters, and pedagogical specialists.

In the category of providing basic tools, the Ministry includes Linux, OpenOffice.org, CMS, LMS as well as some distance learning services. First of all, all new computers in schools, which are co-financed by the Ministry of Education and Sport, have a dual boot for Windows and Linux, and have OpenOffice.org installed for both operating systems. The Linux distribution is called Pingo and is provided in Slovene. Pingo has been developed by a local association called Lugos with the Fedora Linux 3 open source community in Slovenia. For the last two years, the Ministry of Education and Sport, the Ministry of Information Society and the (governmental) Centre for Informatics have financed the localisation in the Slovene language. Currently a tender to cofinance the localisation for the next two years is under preparation.

Secondly, in the area of basic teacher training, among other ICT skills, the programmes include the use of Windows Office as well as OpenOffice.org. The Ministry financed an expert group which supports schools with open source software such as CMS and LMS, and support books have been distributed to schools about the use of Linux and OpenOffice.org. Moreover, the Ministry with National Education Institute and Center for vocational training promotes the use of open source software among teachers, headmasters, and didactic specialists.

As for didactic tools and open content, the Ministry finances teacher training in the area of open source didactic materials (i.e. open content). It has also co-financed new open content didactic material on the web for use in classrooms, with support given for teachers in training to use this material. As well, the Ministry will co-finance, in the future, the creation of didactic material which is not open source, but can be used freely by schools.

To promote the use of open source and open content, the Slovenian Ministry of Education and Sports has started the portal OKO. This project is intended to make the introduction of open source and free educational software into education environments faster and more efficient. The OKO project has started in 2003.

3.5. Ireland: Start Office for all Irish schools

In late 2004, the National Centre for Technology in Education (NCTE), the Irish Government agency established to provide advice, support and information on the use of ICT in education, concluded a licensing and distribution agreement with SUN Microsystems to provide all Irish schools with Star Office, an office suite based on the popular OpenOffice.org. The offer was made to schools in a joint move by the NCTE and SUN Microsystems.

To help schools appreciate the opportunity and to explore the implications of taking up Star Office or substituting the commonly used MS, schools were notified and local information sessions were organised for schools representatives. These sessions were well attended and, following participation, the take up has been significant to date. Schools receive a free CD which allows unlimited copying for staff and students.

Prior to the large scale offer of Star Office, in 2004, the NCTE carried out a number of evaluations of Star Office in a number of schools in order to assess the appropriateness of

this software for schools. The outcomes of these trials proved very positive. Star Office was identified as being a relevant and very useful software tool particularly for schools at primary level.

4. Power Surfing: Xplora, distributing science in its true way - openly

Xplora, the European Science Education Gateway, is operated by European Schoolnet. The Xplora portal is supported by the PENCIL project, a project funded by the European Commission's Directorate General for Research as part of Science and Society.

Xplora offers science teachers tools, information and resources to help them to conduct engaging science lessons that attract students to science. Commonly with 30+ students in classes, science teaching is somewhat blocked by poorly equipped school laboratories. Among the resources that Xplora offers are the usual web based tools like on-line games, downloadable materials and guides to software that is usable in science lessons.

The Xplora portal also offers new tools which have not been used in the classroom before. Among these tools are the web experiments or remote controlled experiments (RCL), in which real experiments are shared via the Internet. Such experiments do not only solve the problem of the true way of science teaching by experiments, but it also opens new pedagogical concepts for science classes. These web experiments deliver results, which students have to process in order to get a lab report.

4.1. Using software – a key skill in scientific research

Participating in science education today means using software as well. Especially for the web experiments, where in most cases students get the result of an experiment as an image, image analysis is a fundamental task. The main tasks for students in science education are: (1) To create a lab report with mathematical expressions, chemical formulas, feynman diagrams, images, tables and graphs; (2) To analyse images, e.g. measure length, angles, area, and intensity; (3) To calculate results, e.g. numerical processing, creating graphs, regression and curve fitting; (4) To create animations; (5) To run simulations; (6) To create and play with mathematical models and (7) To use CAS software to verify the results of calculations.

While office suites text processing software is useful and broadly applicable to be used in schools, it is in many cases not sufficient for specific science tools. One of the examples is simple text editing. For science lab reports, a text writer must be able to handle mathematical equations, chemical formulas and Feynman diagrams, just to mention the most exotic pitfalls.

Many of the the open source software packages have their origins in scientific environments. Thus, there are many applications which can be used for science teaching in classroom with some prior training. Xplora recommends the use of the following software packages displayed in the table (table 1) below for science teaching. On the portal one can find articles and short descriptions for their usage.

<i>Name of software</i>	<i>URL</i>	<i>Description</i>	<i>Application</i>
OpenOffice.org	http://www.openoffice.org	The Open Source office software for scientific text processing, database applications, graphics creation.	Lab reports. Calculation of results Creation of simple charts

<i>Name of software</i>	<i>URL</i>	<i>Description</i>	<i>Application</i>
LyX	http://www.lyx.org	A scientific text processor software, making use of LaTeX properties. Full support of mathematical expressions and all Postscript output from scientific programs.	Lab reports with even the equations, and output of all X11 science software.
Xfig	http://xfig.org	Vector drawing program with a large and extendable parts library.	Preparation of schematic drawings (experimental setups) for lab reports.
Grace	http://plasma-gate.weizmann.ac.il/Grace/	Data analysis program	Plots diagrams of every complexity. Good software for creating regression and line fit.
GIMP	http://www.gimp.org	Graphics program to analyze images	Image analysis (length, angle)
ImageJ	http://rsb.info.nih.gov/ij/	Image analysis program	Analysis of intensity distribution in an image.
Xdrawchem	http://xdrawchem.sourceforge.net/	Program to draw chemical structures	Report on chemistry lab exercises.
OpenRasmol	http://www.openrasmol.org	Program to visualise 3d molecules	Chemistry classroom use and creating images for reports.
Feynman	http://rpmfind.net/linux/RPM/suse/9.0/i386/suse/i586/feynman-1.00-581.i586.html	A program to create Feynman graphs	Particle physics teaching.
Gchemical	http://www.uku.fi/~thassine/ghemical/	A molecular modeling software package	Chemistry teaching in high schools
Gcompris	http://www.ofset.org/gcompris	A software package for the kids	For elementary schools. Many different applications around elementary schools teaching
KDE Edu	http://edu.kde.org/	The KDE Education project	Many educational software packages mainly focusing on lower level education.

Table 1: Open source software for use in science teaching.

4.1 Xplora – Knoppix, making science accessible for schools

To ease some of the organisational problems that schools face in terms of software availability, installation and access in general, the Xplora team developed a live bootable DVD called Xplora – Knoppix. It is based on the Linux Debian distribution and completely contained on a self booting DVD. As this Knoppix version is especially mastered for Xplora, the team has added software applications needed for science education (table 1), as well as a number of educational materials from the Xplora repository. The Xplora – Knoppix release has multilingual support. This concept ensures easy access to scientific tools for education. Being open source software it can be given away freely and copied as many times as needed.

Xplora produces 600 DVDs to be given freely to schools. Moreover, the ISO image of the DVD is freely downloadable from the Internet and can be used to produce the copies needed for the students. Additionally, Xplora has partnered with a company who sells the Xplora – Knoppix DVD for the production plus shipping cost.

Knoppix has proved to be reliable to boot, however for any exceptions there are workarounds described in the trouble shooting section of Xplora.

5. Watching the waves: European Schoolnet's Special Reports

This section gives a short review of Special Reports published by European Schoolnet's Insight Team. Insight is a knowledge base for policy-makers, researchers and practitioners about e-learning in European schools. A series of reports are published in the areas of e-learning policy, innovation and interoperability. Of interest to this paper, some reports are reviewed where the issues touch upon the use of open standards for development of virtual learning environment, a report on open source and open content, and lastly a report highlighting the obstacles that e-learning could face if software patents (i.e. the directive on Computer Implemented Inventions) would pass. These Policy Briefs and Special Insight Reports have attracted some discussion among European Schoolnet's partner organisations, but they have not managed to summon large scale attention or discussion at the European scale among policy-makers.

5.1. Insight Policy Brief: VLEs, Open Standards and Open Source in European Schools

In the series of Insight Policy Briefing, the fourth publication VLEs, Open Standards and Open Source in European Schools came out in late 2003. The brief presented four sections: (1) Importance of VLEs: Views from National Agencies, (2) VLEs and Interoperability Issues, (3) Situation in Schools and (4) Future Developments. The report was based on a survey results on the use of VLEs in European schools, and on the policies of EUN's partners. The reports claimed that when the data was gathered (in late 2002), it appeared that decision-makers and national agencies have taken short term measures to get up and running with VLEs, without considering the adoption of a longer-term e-learning strategy that includes thorough consideration regarding interoperability on a large scale and the use of open standards. The report further proposed that instead of each country continuing developing their separate VLE initiatives, a pooled effort of resources and development skills be used to develop a system that is open and adaptable. Each country could customise the VLE giving it "the look and feel" desired.

5.2. Insight Special Report: Why Europe Needs Free and Open Source Software and Content in schools

In the series of Insight Special Reports, a publication Why Europe Needs Free and Open Source Software and Content in schools came out in March 2004. The report was an attempt to open discussion among European Ministries of Education, national educational agencies, school networks, teachers and other stakeholders. The Special Report brought forward four main areas of consideration and possible collaboration in the area of FOSS and open content: (1) the use of free and open source software in schools; (2) the use of GNU/Linux on servers and

desktops in schools; (3) issues related to the open source development of VLEs, LMSs, LCMSs, etc. and (4) the creation and re-use of "open content".

In addition to above mentioned areas, the report discussed other issues at stake such as the use of free-of-charge educational software, localisation of software for small language groups, greater levels of accessibility, fighting piracy, spending public money wisely and creating new service and local job opportunities. Also, the report covered the area of open educational content offering interesting opportunities for sharing and reusing content.

5.3. Insight Special Report: Software Patents - a potential hindrance of ICT in education

Lastly, at the end of 2004, an Insight Special Report on 'Software Patents - a potential hindrance of ICT in education' came out. The report attempted to open discussion on the European software patent i.e. on the directive on Computer Implemented Inventions. At that moment the directive in question was in a political limbo being ping-ponged between EU institutions. The report attempted to explain the situation at the time with software patents in Europe and compare the situation against the goals for European e-learning in a knowledge-driven society.

Furthermore, the Report on Software Patents examined three main arguments concerning how software patents could potentially harm the European e-learning field if passed. First, the cost of applications could become higher because of the software patent system; the choice of available software could become limited and costs of using underlying communication structures, operating systems and any software could increase. Second, it could have an effect on in-house development of educational applications which seems to be high in European schools⁵³ and still, at the end of 2002, 10 out of 17 Ministries of Education financed in-house VLE development. Third, the roll-out of educational FOSS in schools could be jeopardised by the danger of software patents.

6. Summary

Currently, in the European educational policy and practices landscape, the existing open content and free and open source software initiatives are rather dispersed on a local, national and European level, as well as being spread throughout all educational levels and systems. It is challenging to gain a comprehensive overview of the state of the art, as well as capitalising on the transfer of knowledge gained in one context. However, as this report clearly summarises, European Schoolnet and its members are more and more focusing on the issues around open source and content development. It must be stated, though, that these activities still remain somewhat marginalised in discussions, country reports and conferences, and they rarely receive the limelight that they merit.

The area of open content seems to be rather well accepted concept among EUN's partners. Thus, creating infrastructure and facilitating the content exchange of learning resources in schools is one of EUN's core areas where significant work is conducted to facilitate the co-existence of open and closed content. For example, the implementation of Digital Rights Management Framework is a step towards the co-existence of multiple stakeholders in the field of educational content. Also, some important work will be carried out in the context of CALIBRATE where the development and implementation of an open source collaborative 'learning toolbox' for schools is being undertaken.

What comes to actions taken by European Ministries of Education and other national educational authorities, it seems like they are keen to explore the advantages that open source software and content can offer to education. Interestingly, many have already moved from pilot phases to large-scale implementation. It seems that it would be important to bring these somewhat disparate, but very pertinent national and regional initiatives into the European level to better help the transfer of good practices and to learn from one and other. Furthermore, peer-learning possibilities at the policy level should be better exploited in this area, as has been done in other areas of ICT implementation.

Xplora carries out important work promoting science in education in European schools. The multiple ways to distribute software that is suitable for scientific studies allows schools a better access to the core of science, participate by practicing it.

Finally, the work EUN has carried out in publishing the Insight Special Reports has given a more prominent voice for FOSS in education and been an important source of information for EUN's members and audiences at national levels.

⁵³<http://www.eun.org/goto.cfm?did=25201>

7. References

Links related to section 1:

EUN: <http://www.eun.org> and <http://www.europeanschoolnet.org/>

Links related to section 2:

CELEBRATE: <http://celebrate.eun.org>

CALIBRATE: <http://calibrate.eun.org>

Links related to section 3.1 Belgium:

The Flemish advice on FOSS in education is available in English at

<http://www.ond.vlaanderen.be/ict/english/>

Rortal Klascement: <http://vrijesoftware.klascement.net>

ICT on the menu: <http://www.klascement.net/ictophetmenu>

Digital resources for secondary education: <http://digitaalso.klascement.net>

ENIS: <http://enis.eun.org>

Links related to section 3.2 the Netherlands:

Kennisnet <http://www.kennisnet.nl>

OSS in het onderwijs <http://www.ossinhetonderwijs.nl>

In the section 'voorbeeldprojecten' one can find short descriptions of different case studies

OSSOS, Open Standards and Open Source Software in Government in English

<http://www.ososs.nl/index.jsp?alias=english>

ICT op School <http://www.ictopschool.net>

Dutch association for a wide range of stakeholders in e-learning standards

<http://www.edustandaard.nl/>

Kennisnet on content: <http://contentketen.kennisnet.nl/>

Conference announcement: <http://www.ossos.nl/article.jsp?article=1820>

Links related to section 3.3 Estonia:

Distribution based on Mandriva (Mandrake) developed by an NGO called Offline.ee and supported by Tiger

Leap Foundation, expected release fall 05.

Briefing on open parental code software use in school (12/04/2004):

http://www.tiigrihype.ee/eng/arhiiv_1.php?uID=49

VIKO is released under GPL in 2003. It is a simple PHP/MySQL based system, currently available only in

Estonian. The development continues by user community and there are plans to make it possible to localise

the system. <http://www.htk.tlu.ee/viko/>

IVA <http://www.htk.tpu.ee/iva/>

Papers:

Laanpere, M., Põldoja, H., Kikkas, K. (2004). The Second Thoughts about Pedagogical Neutrality of LMS's. In: The 4th IEEE International Conference on Advanced Learning Technologies. Joensuu, Finland. 30

August - 1 September 2004. Los Alamitos: IEEE, 807-809.

Põldoja, H. (2003). Learning Management System IVA. e-Learning in Science and Environmental Education.

In: Proceedings of the International Conference October 1-4, 2003, Tartu, Estonia. Tartu:

University of Tartu, 52-54. <http://www.ut.ee/eLSEECConf/Kogumik/Poldoja.pdf>

Põldoja, H., Laanpere, M. (2004). IVA - Learning Management System based on Fle3. [Unpublished].

http://www.hanspoldoja.net/work/pdf/iva_flebook_article.pdf

Blog on free and open content in education: Vabavere <http://www.htk.tlu.ee/vabavere/>

Links related to section 3.4 the UK:

Open Source Software in Schools: A study of the spectrum of use and related ICT infrastructure costs – Project report

http://www.becta.org.uk/corporate/publications/documents/BEC5606_Full_report18.pdf

Open Source Software in Schools: A case study report

http://www.becta.org.uk/corporate/publications/documents/BEC5606_Case_Study_16.pdf

Using Open Source Software in Schools: Information sheet

http://www.becta.org.uk/publications/documents/BEC5606_Information_Sheetrev.pdf

Previous UK Government studies include: Office of Government Commerce [2004] "Open Source Software

Trials in Government: Final report" <http://www.ogc.gov.uk>

Links related to section 3.5 Slovenia:

OKO <http://oko.edus.si>

Links related to section 3.6 Ireland:

Irish news on Star Office <http://www.ncte.ie/NewsandEvents/Newsletter/d2413.HTML.html>

and a press release - in pdf-

format:<http://www.ncte.ie/documents/pressreleaseforStaroffice.pdf>

Links related to section 4

Xplora: <http://www.xplora.org>

Web experiments:

http://www.xplora.org/ww/en/pub/xplora/megalab/web_experiments.htm

GI-Knoppix, a predecessor of Xplora-Knoppix:

http://www.xplora.org/ww/en/pub/xplora/library/software/gi_knoppix___software_for_sci_e.htm

Linux-cd.info: <http://linux-cd.info/>

Links related to section 5

Insight: <http://insight.eun.org>

Insight Special Reports <http://insight.eun.org/ww/en/pub/insight/misc/specialreports.htm>

Acknowledgements

The report is part of European Schoolnet's Insight Special Report services. The document will be found at <http://insight.eun.org/ww/en/pub/insight/misc/specialreports.htm>

The country briefings included for the report are based on EUN's members contributions, thanks to Jan de Craemer from the Education department of the Ministry of the Flemish Community, Hans Poldoja from Tallinn University, Piet Hein Minneché, the programme manager of OSS in het onderwijs in the Netherlands, Borut Campelj from Slovenian Ministry of education and sport, Jerome Morrissey, the Director of Irish National Centre for Technology in Education.

Helping to develop an Open Source Curriculum: the case of TOSSAD E.U. funded project

The TOSSAD WP4 Team: Graham Attwell, Stephen Barrett, Kaan Erkan, Selahattin Kuru, Michele Marchesi, Enn Öunapuu, Oleksandr Ulybin

contact author: Michele Marchesi, DIEE – University of Cagliari
michele@diee.unica.it

Abstract

In this paper we present the aims and the activities which are being carried on in the context of TOSSAD project, to help developing an Open Source Curriculum. TOSSAD (Towards Open Source Software Adoption and Dissemination) is a Coordination Action funded in the context of 6th E.U. Framework Programme. The main objective of the project is to start integrating and exploiting already formed methodologies, strategies, skills and technologies in F/OSS domain in order to help governmental bodies, educational institutions and SMEs to share research results, establish synergies, build partnerships and innovate in an enlarged Europe. One of the workpackages of the project, WP4, aims to develop an OS curriculum. The initial attention is directed to courses that could be offered in a high school and university computer science/computer engineering departments. The workgroup is discussing and elaborating on the content inventory and is proposing various course topics and a master curriculum. The whole activity will be performed strictly cooperating with similar initiatives and projects undertaken in Europe. The paper describes and discusses in detail the ideas and experiences of TOSSAD WP4.

Introduction

In Europe, many think that it would be desirable to improve the usage of Free, Open Source Software (F/OSS) in all branches of IT and public life, in general. Although there is a committed open source community in IT-strong countries of Europe, there is much more to be done. F/OSS communities throughout Europe can achieve better results through co-ordination of their research activities/programmes that reflect the current state-of-the-art.

The TOSSAD project (Towards Open Source Software Adoption and Dissemination) has been proposed and funded by European Union with the aim to start integrating and exploiting already formed methodologies, strategies, skills and technologies in F/OSS domain in order to help governmental bodies, educational institutions and SMEs to share research results, establish synergies, build partnerships and innovate in an enlarged Europe.

More precisely, the TOSSAD project aims at improving the outcomes of the F/OSS communities throughout Europe through supporting the coordination and networking of these communities by means of state-of-the-art studies, national program initiations, usability cases, curriculum development and the development of collaborative information e-bays and web-based groupware. By conducting these actions on an international European level, with inclusion of the ACC and NMS countries, the TOSSAD project will create sufficient momentum for a general acceptance and coordinated boost of F/OSS development.

The project is in line with the EU policies that encourage wider usage of F/OSS in Europe on every platform. It also supports the objectives of the IST Work Programme 2003-2004, in particular, as stated clearly in Part 2.2.2: "The development of open standards and open source software will be encouraged when appropriate to ensure interoperability of solutions and further innovation."

The detailed objectives of the project are:

- To give a clear picture of the current status of F/OSS in Europe and explain the main reasons for technical and social barriers against its wider deployment.
- To build the basis to start up national programmes for improved usage of F/OSS in "F/OSS target countries". This objective's outcome will be measured by the quality

and the quantity of the intentions and initiatives of governmental bodies through adoption of F/OSS with concrete and applied national plans.

- To integrate current applied research activities in Europe in the field of usability and accessibility. A success factor for this goal will be how much the outcomes generated from usability tests and surveys will support today's technological needs.
- To develop and revise F/OSS training standards in intermediary education, general public schools and universities, and help form the infrastructure and standardization of training documents.
- To disseminate the results of all project activities, producing the dissemination plan, running the project's web site, forming the brochures and other instruments (i.e. newsletter), and showing up in various media for increased awareness.

More specifically, this paper presents in detail Workpackage 4 of TOSSAD project, which is related to F/OSS education and curriculum development.

The Workpackage 4: The F/OSS Curriculum Development

In order to generate a stable and viable economy based upon local developers leveraging F/OSS to their advantage, the educational aspect is very important indeed. Without it, the less developed countries run the risk of falling behind in the technology race.

It is important to spread understanding how F/OSS culture can be an attractive alternative in terms of cost, quality, reliability, security of software solutions, and how it is invaluable in terms of community, democratisation, and human-rights. Quoting an observation of the European Commission Working Group on F/OSS, "Consider the recommendations not as 'how to help open source software', but 'how to help Europe to benefit from open source software', one can only restate that if F/OSS can be relied upon to help Europe, it can help everyone else; and this lesson must start from education".

A mass of trained IT professionals is a very important factor impeding the spread of F/OSS in many developing countries. This issue can be best addressed by taking a second look at the educational and vocational training policies, which should make sure that the students get a chance to know multiple technologies, and are not limited by the predominance of a single vendor or technology in the educational curriculum and in laboratories ("Free as in Education: Significance of the Free/Libre and Open Source Software for Developing Countries", Niranjana Rajani, <http://www.maailma.kaapeli.fi/FLOSSReport1.0.html>)

The relationships between F/OSS and education, however, are multi-faceted, and some of them are yet to be considered in depth. Among others, we may report the following issues:

- F/OSS software needs education and training, as any software package. If you don't know what the software does, and how to use it, you cannot take advantage of it. One needs an educated section of the population to fulfill the full potential of F/OSS
- Learning F/OSS systems poses new issues, because true mastering of a F/OSS application involves cooperating with the community of its developers. This in turn leverages the knowledge of the application.
- F/OSS helps, enhances, and complements education by providing tools to promote education. This is achieved on one side because using F/OSS enables schools, especially in developing countries, to set up computer laboratories at a lower hardware cost, and at no software cost; and on the other side because exposing students to F/OSS, a working software that can be read, studied and modified in cooperation with its authors, can be a very powerful tool for achieving well-educated professionals, able to substantially advance their economy. Moreover, as quoted above, F/OSS culture it is invaluable in terms of community, democratisation, and human-rights.

For all the above reasons, studying and developing F/OSS education is very important. However, F/OSS education is in turn a multi-faceted concept. Many kinds of F/OSS curricula and courses may be devised, at various levels, very different from each other:

- Courses and curricula about using the most popular F/OSS desktop applications — F/OSS office automation software, mail applications, Web browsers, Wiki's, etc. — even on proprietary operating systems.
- Courses and curricula about F/OSS server application & management – Linux operating system, application server (Tomcat), Web server (Apache), databases, middleware, and related system applications.
- Courses and curricula about F/OSS software development tools – IDE (Eclipse), Versioning Systems and related tools.
- Courses and curricula about how to develop and take advantage of F/OSS software — the software engineering of F/OSS. They are related to ongoing research on methodologies and tools for F/OSS development, and aim to train software developers able to build, customize and consult on F/OSS applications, being active members of the F/OSS development community.
- Use of F/OSS software in computer science courses and curricula, as a cheap and powerful mean to help understanding the computer science concepts.
- Courses and curricula about evaluating the economic impact of F/OSS adoption, and about the F/OSS business model for software firms. These issues are again related to ongoing research.

There is a recurring request for help in preparing a series of courses on open source software in general. Although curriculum development lags behind the progress of F/OSS, we believe it's time to begin working on F/OSS curriculum development, because a) there is now sufficient knowledge and skills, b) there is increasing interest c) the number of courses in F/OSS and Linux have grown rapidly during the last two years.

The workgroup of WP4 gathers partners with deep and complementary knowledge in software engineering, university curricula development, e-learning and collaborative learning, application of open source methodology and business models to real world problems. The academic partners are: University of Cagliari (Italy), Technical University of Tallinn (Estonia), and Trinity College (Ireland); the industrial partners are IOTA (Turkey), The KnowNet (U.K.), and Ukrainian Lviv Institute for Business Informatics (Ukraine). WP4 partners are working together in order to define one or more broadly accepted, detailed curricula for F/OSS. We will focus in particular on courses and curricula about F/OSS operating system Linux related system applications, and courses and and F/OSS software development tools, not excluding studying and giving suggestions on other education-related subjects.

The initial attention is directed to courses that could be offered in a high school and university computer science/computer engineering departments. This because we believe that they are a key factor for F/OSS success, being today's students tomorrow's professionals. Moreover, convincing schools of the goodness of the F/OSS approach, and of the fact that it is fruitful to invest in it, is perhaps easier than convincing end-users organisations or firms. The workgroup will discuss and elaborate on the content inventory and propose various course topics and corresponding curricula.

This curriculum planning will provide a solid foundation for students in high school with grade 10-12 and university students (year 1-4). The workgroup includes technology coordinators, curriculum planners, instructional technologists, teachers and students with technical training in the Linux operating system. It will support and disseminate the utilization of F/OSS technology in education and promote the critical technology and career competencies that computer industry increasingly demands.

The project plans to strengthen the relationship between educational institutes and the business community by organizing events between industry mentors, students, technology executives and teachers. This because it would not be sensible to develop a curriculum "in vacuum", but strong collaboration with software industry is of the utmost importance for the success of the workpackage. Additionally, especially university students will benefit from different aspects and philosophies of technology planning and development with the help of this curriculum.

Work carried on in this workpackage includes also additional research and collaboration activity with various projects that are trying to push F/OSS technologies within schools, in particular with Special Interest Group in Open Source Software for Education in Europe (E.U. project SIGOSSEE www.ossite.org), SkoleLinux www.skolelinux.org/portal, KDE edutainment edu.kde.org and DebianEdu. With this target in mind, the adoption and acceptance of a F/OSS curriculum will definitely be easier in an enlarged Europe.

In the first six months of the project, various curriculum plans worldwide have been examined; the remaining months will be used to develop a F/OSS curriculum, tutorial and methodology for high schools and universities. The stakeholders of this workpackage will gather twice during the lifetime of the project. A key output of WP4 will be the "F/OSS in education" training program, a 1-day event covering the use of F/OSS in high schools and universities and targeting education authorities and institutes, described in the section below.

As said before, one problem that may exist in setting up a curriculum based on F/OSS is the sheer diversity of options available. The workpackage partners are also debating on the critical question: "Should the curriculum attempt a broad overview of everything, or should it allow potential students to concentrate upon topics of their own choice?" For example, a broad range would qualify someone's ability to install Linux, write simple applications in a scripting language, and deal with a range of everyday problems. On the other hand, students may have the need to install Linux on a variety of hardware, make it as secure as possible, and administer the system for a complex organisation. The obvious solution would be to have a module-based curriculum whereby students could attempt the modules of interest to themselves. The curriculum and its courses will target teaching F/OSS and F/OSS development. It will be possible also to propose new versions of traditional computer science courses, having F/OSS at course topic level.

The curricula and course definitions produced by WP4 will in turn be made available through Creative Commons license for the content, in the true F/OSS spirit. So, the results of WP4 will be completely open themselves.

F/OSS in education training program

"F/OSS in Education" is a 1-day training session and a final birds of a feather session (BOF) covering the use of F/OSS in high schools and universities. This training program targets local authorities, education and technology ministries and all kinds of educational institutes and focus on the following issues:

- Syllabus preparation
- Curriculum development
- Educational course content
- Course evaluation mechanisms
- F/OSS e-learning methodologies

Workpackage 4 Tasks

The F/OSS Curriculum Development workpackage is as usual divided in tasks. These tasks are:

- Organizing two workshops and the training program on “F/OSS in education”; the first workshop will study and discuss the various curriculum plans worldwide, and the second will discuss the developed curriculum. The workshops will be held together with other TOSSAD and/or F/OSS events.
- Evaluating of the needs of industry to enhance the qualifications in F/OSS, based on the societal objectives to improve the F/OSS adoption. Local industries, including SMEs, will be contacted and interviewed, in order to assess their orientation toward F/OSS and to define which skills are needed by them to use F/OSS. To this purpose, a questionnaire has been developed to help assessing industry needs. The results of this task will be used in developing F/OSS curricula and courses.
- Determining best F/OSS training methods, whether these are classroom based learning or e-learning (distance learning). This task will be developed by surveying and recommending pedagogic processes and methodologies for training in F/OSS. These recommendations will inform the nature and design of the proposed standards and curriculum.
- Developing and revising F/OSS education and training standards and helping form the infrastructure and standardization of training documents. This task will build on the results of previous tasks, and will produce one or more proposed curricula and course programs.
- Exchanging of information about curricular aspects of F/OSS and their further development

Conclusions

The educational aspect is crucial for spreading and acceptance of F/OSS among European public administrations, firms and organisations. To this purpose we need teachers, teaching material, courses at various level, entire curricula. Many F/OSS projects and organisations have realized this, and are actively working to produce the needed competencies and material. To be effective, however, we need coordination among efforts, and the definition and acceptance of course syllabi and curricula in high schools and universities.

The TOSSAD project aims to help the definition of such syllabi and curricula, gathering information on present efforts, making proposals to competent authorities, and disseminating knowledge about the issues found and the proposed solutions. To this purpose, TOSSAD partners are looking for contributions, will strictly cooperate with existing initiatives, and will organize dissemination events.

Building Eclectic Personal Learning Landscapes with Open Source Tools

Marco Kalz, M.A.
Educational Technology Unit, Fernuniversität in Hagen
marco.kalz@fernuni-hagen.de

Abstract

There is an ongoing trend towards modularization of Learning Management Systems and other E-Learning Applications. Modularization should add flexibility to the previously static environments that have been used for e-learning. This trend concerns both commercial LMS vendors (e.g. the concept of building blocks in Blackboard) and open source developers (e.g. Moodle). Based on the model of the personal learning landscape (see Tosh/ Werdmuller 2004) this paper describes another approach to develop personalized learning environments. Through an eclectic use and combination of different systems and services this paper demonstrates how to develop personalized learning environments with the combination of different open source applications. Software for static content (Mambo CMS) is combined with dynamic systems like b2evolution and Mediawiki. After "dancing the mambo", "doing the evolution" and "hopping on the wiki bus", all systems are connected through the use of the "magic glue", RSS. Different use cases and a development outlook regarding inter-system development needs are provided.

1. Flexibility and personalization through modularization?

Our Western Society is making a shift to a knowledge society where static knowledge is not important for competency development and job success but rather access to and renewal of knowledge. Universities have not really targeted this changing way of knowledge renewal as the outcomes and the curriculum are still the centre of most academic assessment. Besides the certificates and finished papers, students in Universities are learning much about learning. Through the 'product-centred' approach of many course programmes students learn to set the focus on the product and not their learning process. This is an unintended and false kind of meta-learning. The learning process is in this sense only the way to the product and has no real value in itself. This trend is also mirrored in the tools that have been recently used in academic e-Learning efforts. Many Learning Management Systems (LMS) have been implemented in universities in Europe, the main task of which is the delivery of learning content and the administration of learners. These systems have been criticised from different e-learning practitioners and scientists because of their limitations in learning action, reflection and communication opportunities (Siemens 2004, Morrison 2004, Schneider 2005). They are mainly useful for passive consumption of presented content and they are not very flexible for different learning activities and learning connected to the life of the students. Schneider coins them as supporting a "transmissive pedagogy" (Schneider 2005).

To overcome these limitations there was and is an ongoing effort to build e-learning systems through a modular approach to develop "personalized learning environments". This trend seems to influence both commercial and proprietary system development (e.g. the concept of Building Blocks in Blackboard) or open source products such as Moodle or Atutor. Modularization addresses two main problems that most e-learning environments still lack. There is less flexibility in functionality and activity possibilities and there is very less scalability in the user interface. Both aspects address the problem of user demands and personalization. But the question remains as to how useful these personalization efforts are, when personalization is only the choice of colours and shortcuts to favourite learning objects. Another issue is flexibility for the future. If you build e-learning environments from one vendor or on just one open source product you are either dependent on the development plans of the vendor or on an 'autopoietic' open source community. Morrison keeps warning us against possible "lock-in" if universities have the intention to build a monoculture of tools that is based on just the one-and-only Learning Management System (Morrison 2005). The modularization concept does not really solve the problem of offering

flexible and unique learning environments because it is too narrow. An alternative to this kind of infrastructure for learning and competency development is the concept of the personal learning landscape. The personal learning landscape is a special kind of e-portfolio which I will introduce in the following chapter.

2. E-portfolios and the Personal Learning Landscape

The portfolio concept is not a new concept. Indeed the French teacher, Celestine Freinet, introduced them in the late twenties of the last century in his classes. In the last years there appears to be a rebirth of this concept – mainly driven by technological development. According to the European Institute for E-Learning (EifEL), every citizen in the European Union should have his own e-portfolio by the year 2010. The electronic portfolio (e-portfolio) can be understood as a “a collection of authentic and diverse evidence, drawn from a larger archive representing what a person or organization has learned over time on which the person or organization has reflected, and designed for presentation to one or more audiences for a particular rhetorical purpose” (Educause Learning Initiative 2003). Although research into electronic portfolios has a short history, there are already two development directions for the e-portfolio-concept: “The 'e-portfolio' used for final assessment/ job seeking where the emphasis is on the product(s) and then the 'e-portfolio' used for reflection, deep learning, knowledge growth and social interaction where the emphasis lies on the process” (Tosh/Werdmuller 2004, 2). They call the second kind of e-portfolio a “personal learning landscape”. Helen Barret differentiates three directions and audiences for the use of an e-portfolio: portfolios for accountability, portfolios for marketing and portfolios for learning. Portfolios for accountability are product centered and have the main task to “document and assess the achievement of externally defined skills or competencies...students usually view this type of portfolio as something 'done to them' rather than something they WANT to maintain as a lifelong learning tool” (Barret/Carney 2005). Portfolios for marketing are tools for self-marketing of job-seekers. Their aim is to present the applicant in the best position achievable. Portfolios for learning are – according to the authors – based upon a constructivist model. The emphasis of this kind of portfolio is on the individual learning process, reflection and new plans for learning. Attwell identifies seven different functions of an e-portfolio for learning (Attwell 2005):

1. Recognizing Learning

Learning in a formal environment is usually recognized when pre-specified products are achieved. e-portfolios can be a means to recognize smaller learning achievements.

2. Recording Learning

E-portfolios can be containers for recording formal assessment through scanned certificates for example. Additionally they can be used to record informal learning activities.

3. Reflecting on Learning

Reflection is an important part of a learning process. The e-portfolio can be used for private, semi-public or public reflection of this process.

4. Validating Learning

Validation in e-portfolios can be a self-validation or a validation from other persons. Validation means to „proof“ that learning has happened. This validation can have different forms and can appear in different media.

5. Presenting Learning

The presentation of learning is important in e-portfolios. This presentation can be used for job application or for academic application. Due to the importance of lifelong learning this presentation can change over time.

6. Planning Learning

The learning process can be planned with the help of the e-portfolio. The learner can view his personal learning history through his e-portfolio and can view his next steps in personal competency development.

7. Assessing Learning

Assessing means external control and judgement over the learners achievements.

All these aspects of e-portfolios are important for the learning process and it seems to make sense to develop applications which support them. Tosh and Werdmuller developed an application that stresses reflection and social networking between learners: Elgg is a very flexible tool that supports personal weblogs, tagging and social networking in one application. Although the software is still in a beta stage, it has a maturity that is impressive. But, there are other options to design personal learning landscapes. Instead of building new applications from scratch, I think it makes sense to concentrate in the future on systematic combinations of existing Open Source tools for learning and competency development. The example of the XAMPP-Server that is used in many schools and universities has shown that there is an emergent output from this approach. The XAMPP team did not build a new application, but combined existing solutions (Apache, PHP, MySQL) which are all hard to install and configure for non-technical experienced teachers and users. So they built an easy installation package for different platforms in which these applications are combined. The following concept is based on such ideas of emergent usefulness through combination of existing tools. In the following part I will introduce a personal learning landscape that is built with the help of three different open source applications.

3. Dance the Mambo, do the evolution and hop on the Wiki bus – and don't forget the glue

The personal learning landscape is a special type of e-portfolio that stresses the importance of the learning process and serves as a framework to integrate different learning activities. In this section I provide an example of the conception and implementation of an e-portfolio/personal learning landscape, which is a combination of different open source tools. These tools should support the above mentioned functions of an e-portfolio for learning. In general the personal learning landscape should support and combine the following main tasks: 1.) Presenting static content in a professional online environment where updates are easy to alter, 2.) Possibilities for reflection and peer-discussion for academic and private subjects and 3.) a place for living documents that grow during the academic career. To support these tasks, three open source Content Management Solutions have been selected and should be combined to a coherent personal learning landscape.

3.1 Dance the Mambo

The first thing we need for a personal learning landscape is a place to present static content. You need static pages for the things in life that do not change that often like your CV or diplomas and references. There are a huge number of Content Management Systems to support this part of the learning landscape (Baumgartner & Kalz 2004). In this case I have chosen a system called 'Mamboserver' because of its ease of use and rapid development speed and because of its great community⁵⁴. Mamboserver is a CMS based on PHP/MySQL that is very easy to install and administer. Nonetheless it is very flexible and it can be taken as a framework to integrate other different tools and services. Mambo has different content types: It can support static content pages and it supports the integration of RSS Feeds as an own content type. Regarding our personal learning landscape, it is very useful that it supports two different WYSIWYG editors in all content pages, because changing content is easier through these editors than using HTML. As mentioned, Mambo should be used to host the static content for the personal learning landscape. So every student can write their C.V. in Mambo, upload scanned references and offer a central contact page. If these static contents should be visible only to specific users, this can be done through the groups and access system that can be easily modified. Mamboserver menus are very easy to customize and different free skins/templates are offered. According to the different functions regarding e-portfolios for learning, the Mamboserver can be used to fulfill different aspects. Its main function is the recording of learning and the presenting of learning. Additionally, the Mamboserver serves as a 'meta-container' to integrate other tools and services, as I will show later.

⁵⁴Meanwhile the most Mamboserver developers have decided to do a fork that is called Joomla.

3.2 Do the evolution

After having implemented the frontdoor of the personal learning landscape, we have to think about a place for different dynamic content. There has to be a place for (peer) communication, inquiry and reflection in our personal learning landscape. Thoughts are preliminary in the backdoor of the personal learning landscape and peer communication is important to get feedback. A weblog system seems to be suited very well to support these actions in the learning process. There has been an intensive discussion on Multi-User/Multi-Blog-Platforms in the last months (Farmer 2004, Levine 2005). Regarding Open Source blogging platforms, Wordpress and B2evolution are two Open Source systems that seem to have the maturity to be suited as a bloghosting basis for many users. Because of some problems with the latest release of Wordpress MU and different personal experiences, b2evolution is chosen as the basis for hosted blogging in the personal learning landscape. B2evolution is a multi-user-multi-weblog application with an easy administration and blogging interface. The software is available in 12 different languages and there is a great community working on support, documentation and development. There are different standard functionalities and additional hacks to protect the weblogs from spam. The spam blacklist is a very intelligent solution: if there is comment spam on a b2evolution-weblog the author can delete this spam with one click and report this spammer to the central spam blacklist on the b2evolution nserver. So all b2evolution blogs share their anti comment spam blacklist. At the moment, the administrator has to update his local list regularly, but automated solutions are prepared for the next releases. There is additionally a possibility to minimize referer-spam which is another problem for weblogs. To integrate weblogs successfully in an academic environment there are two important requirements: The first one is the autocreation of weblogs and the other is LDAP support. B2evolution is at the time in the standard release not suited for the autocreation of weblogs but a small hack can add this functionality. Additionally the next release will have LDAP Support (see the paper of Michael Klebl in this Proceedings) to authenticate students. These aspects have been discussed intensively in the Edublogger Community (Farmer 2005, Norman 2005). For communication purposes, comments and trackbacks are implemented. There are around 30 skins available that can be modified by the students with a little knowledge of HTML/PHP. Blogging can be used in many different instructional ways – in the case of the personal learning landscape the blogging application is used for reflecting on learning and planning on learning. Blogging as personal public reflection is a good possibility for peer communication with other students and teachers, regardless of their physical presence.

3.3 Hop on the Wiki Bus

So after we have added a weblog for reflection and peer-communication to the personal learning landscape, we wish to offer a place for the unfinished texts and the preliminary writings that happen very often in academic education. Because of its development speed and its usability, Wikipedia's Mediawiki is chosen as the solution for this, but there are many other wiki clones that could be useful. Mediawiki is developed and tested by the huge Wikipedia community. It is a Wiki system based on Php and MySQL that is very flexible in configuration and adaptation. Mediawiki has a basic WYSIWYG editor for fast editing of texts. The Wiki can be configured for open editing, for members editing or completely closed. These different access options can be important if students want to have some privacy in their personal wiki. The Wiki can be used for recognising learning, planning learning and assessing learning. Besides the use of a Wiki for preliminary versions of learning outcomes (papers etc.), the Wiki can be used for knowledge management. So the student can use the Wiki to structure a new 'knowledge field' with the help of a Wiki. In this mode, small learning efforts can be recognized in a Wiki and different structures can be easily managed through living documents. Teachers and tutors can have access to these living documents and they can add comments to the students personal knowledge structures.

3.4 Fix it with RSS-Glue and add some Fun and Games

Now that we have built a personal learning landscape out of three different open source applications, we can connect them through the magic glue of RSS. RSS (translated as Rich Site Syndication or sometimes Really Simple Syndication) is a standard protocol for content syndication (Richardson 2004). Mamboserver has an integrated RSS interpreter so that we can add the RSS feeds from our dynamic applications in Mamboserver. There are many different possibilities to add RSS Feeds and third-party services to our personal learning landscape. We can integrate the latest posting on the students weblog on the frontpage of

Mambo. Additionally we can add the latest changes on his personal wiki implementation and show his last edits. We can add a newsreader page in the menu on which we can show interesting feeds for the student. To offer a little more personalization, students can integrate different third-party services to their learning landscapes. Take social bookmark managers like Furl, Del.icio.us or maybe Citeulike, for example. If the student is using one of these services for inquiry his latest additions can be easily integrated in Mambo through RSS/Javascript. So they can maybe find other interesting sources for inquiry through their personal profiles. But, there are several more examples with which the students can personalize and modify their learning landscape. Take a service like RSS Weather or the RSS Calendar. Adding local weather to the personal learning landscape can connect the students 'virtual' presence with their 'physical' presence. There are many more examples that can be integrated in the personal learning landscapes. To reach a real **personal** learning landscape, these additional 'goodies' are very important because they can be motivating for students and they can help them to modify and change the style of their landscapes.

4. Inter system development needs and problems

There are some problems connected with the proposed concept that have to be addressed. The first thing is the administration effort. It is much more work to administer and update three applications for every student than to manage just one. The effort of installation is high if you have to install three systems for every student. So automatic installation scripts have to be developed that integrate all tools into one installation process. The same problem is with the databases. To integrate the tool in a personal learning landscape you have to administrate three different databases for every student. But, there are already integration efforts for Mambo/Mediawiki and Mambo/b2evolution so further development could build on these efforts. Another issue is a help system that supports students in configuring and modifying their personal learning landscapes.

5. Take open source for education to a new level: Rip, Mix and Burn

As I have shown in this paper, there is potential in combining different Open Source tools to build valuable applications for learning, inquiry and assessment. Two different ways to implement e-portfolios have been identified and the idea of the personal learning landscape seems to be suited for a focus shift from outcomes to processes of learning in universities. With a small amount of work, the proposed concept should be realized in the future in a way the XAMPP project has done it. They brought a really important added value for non-technical, experienced educators by combining everything you need to run a webserver (Apache, MySQL, PHP) and integrate it in an easy-to-install-application. My conclusion for the Open Source Movement in education is that we have to concentrate on the next development level of Open Source Software. This is the 'Rip, Mix and Burn' opportunity that is the most important difference between Open Source and proprietary software. The most important issue is not about pricing and cost of ownership but it is simply about freedom and flexibility.

Reference

- Attwell, Graham (2005): Recognising Learning: Educational and pedagogic issues in e-Portfolios. Retrieved September 23rd from http://www.knownet.com/writing/weblogs/Graham_Attwell/entries/5565143946
- Barret, Helen; Carney Joanne (2005): Conflicting Paradigms and Competing Purposes in Electronic Portfolio Development. Submitted to *Educational Assessment*, an LEA Journal, for an issue focusing on Assessing Technology Competencies.
- Baumgartner, Peter & Kalz, Marco (2004): Content Management Systeme aus bildungstechnologischer Sicht. In: Baumgartner, Peter; Häfele, Hartmut & Maier-Häfele, Kornelia: Content Management Systeme für E-Education. Innsbruck.
- Educause Learning Initiative (2003): E-Portfolios.
- Farmer, James (2005): Multi User Weblogging – What are your options. Retrieved September 12th 2005 from <http://incsub.org/blog/?p=266>
- Morrison, Derek: E-learning Flexible Frameworks and Tools: Is it too late? The Director's Cut. In: Proceedings of ALT-C, Exeter, November 2004. Retrieved September 12th 2005 from <http://www.bath.ac.uk/e-learning/download/DM20040909.pdf>.
- Norman, D'Arcy (2005): Massively Multi User Weblogging. Retrieved September 12th 2005 from <http://www.darcynorman.net/2005/02/20/massively-multi-user-weblogging>.
- Richardson, Will: RSS – A quick start guide for educators. Retrieved September 12th 2005 from <http://static.hcrhs.k12.nj.us/gems/tech/RSSFAQ4.pdf>
- Schneider, Daniel (2005): Kollaborative Webportale in der projektorientierten Ausbildung. Lecture at Learntec 2005. Retrieved September 12th 2005 from <http://tecfa.unige.ch/proj/seed/catalog/docs/learntec05.pdf>
- Siemens, George (2004): Learning Management Systems: The Wrong Place to Start Learning. Retrieved September 12th 2005 from <http://www.elearnspace.org/Articles/lms.htm>
- Tosh, David & Werdmuller, Ben (2005): Creation of a Learning Landscape: weblogging and social networking in the context of e-portfolios. Retrieved 12th September 2005 from http://www.eradc.org/papers/Learning_landscape.pdf

Adapting Open Source Software for Education: Challenges, methodologies and results

Jim Henderson
Learning and Teaching Scotland
j.henderson@ltscotland.org.uk

Evan Brown
Cànan Ltd
evan@canan.co.uk

Chris Mitchel
Cànan Ltd
chris@canan.co.uk

Abstract

Two recent projects, both commissioned by Learning and Teaching Scotland on behalf of the Scottish Executive and developed by Cànan Ltd, employ open source software to deliver educational resources. The end products address very different educational requirements but both were made possible by the implementation and adaptation of open source software. This paper will examine the rationale for using an open source development model from the public policy and technical points of view, firstly in the production of OpenOffice.org Gaelic, a Scottish Gaelic localization of the OpenOffice.org productivity suite, and also in PhraseBox, a forthcoming corpus search engine that is designed to enrich and support teaching of English in the 3rd and 4th years of secondary schooling in Scotland. We will examine the different challenges presented by the two projects and ways in which the open source community helped to provide solutions. OpenOffice.org Gaelic is the first industrial strength software application to be made available to students of Scottish Gaelic, a language spoken by fewer than 60,000 people. PhraseBox, by contrast, is a large scale phraseology search engine, inspired by academic research, which will be piloted initially in Scottish schools before being applied elsewhere. Finally, we consider development and implementation methodologies that enable the successful adoption of products derived from open source software within different communities.

Open Source for the curriculum

The use of open source software in front-line education is not a new idea. In 1998, the Mexican government announced that it would install the Linux operating system on 140,000 personal computers in elementary schools as part of the ScholarNet⁵⁵ program. This project provides Mexican students with access to the web, e-mail and desktop applications like a word processor and spreadsheet. Coordinators cite the high costs of licensing and ownership that are typically associated with commercial products as the prime motivation for deploying open source software (OSS). In addition to the operating system, it was found that software such as the Mozilla web browser, the Thunderbird e-mail client and the OpenOffice.org productivity suite offer a platform of tools with greater functionality and flexibility than would be financially affordable using proprietary products. In this case, restricted resources necessitated a move to open source software, but the decision has since been vindicated on both a technical and policy level. ScholarNet benefits from widespread support and the technical expertise contributed by members of the open source community, thus harnessing an international network of product specialists that few commercial companies can sustain. Savings on license costs enable a program of ongoing support and platform development which would not otherwise be possible. Finally, it has been noted that the robust foundation provided by Linux allows the service to run effectively in circumstances where budgets for maintenance and

55 Kahney, L., Mexican schools embrace Linux. In: *Wired Magazine*, November 6, 1998. Also at: <http://www.wired.com/news/technology/0,1282,16107,00.html>.

modernization of existing hardware are limited.

The story of ScholarNet highlights many of the traditional justifications for and benefits flowing from the adoption of open source software in an educational environment. Within this domain, however, a distinction can be drawn which serves to present even greater opportunities for the integration of OSS projects, methodologies and expertise in educational software, online environments and learning platforms. It is clear that the open source community makes an increasingly large and important contribution to the development of education systems worldwide. Recent studies demonstrate that there is a growing institutional acceptance of the open source development and procurement model. It is suggested, however, that many of the products currently used to support learning and teaching are focused rather narrowly in the field of technical service delivery and that there has traditionally been less emphasis on harnessing such resources to deliver educational tools and materials that directly develop student knowledge and understanding. The two projects presented in this paper sit on either side of the divide that we perceive. They indicate two different strands for educational OSS development which, it is hoped, will both continue to mature in the future.

This paper and the conclusions drawn herein are derived from experiences in educational provision at the Primary and Secondary school level in Scotland. The mantra that using open source is better because it costs less is important. However, it completely overlooks other advantages that are of much greater benefit in an educational setting. A focus on the financial imperative is understandable but recent research by an educational body in the United Kingdom confirms the importance of other considerations. The British Educational Communications and Technology Agency (BECTA)⁵⁶ has a remit to support all four UK Education Departments in developing sustainable plans for the integration and deployment of information technology. It published a report in May 2005, following a study of the adoption and use of open source products in a number of schools. The stated aims of this study were: to examine the effectiveness of 'the open source software approach' in the delivery of the school curriculum; to examine the total cost of ownership, including ongoing support costs, inherent in the use of commercial products as opposed to open source software and to highlight examples of successful OSS usage in British education⁵⁷.

In general, open source software is used by members of the BECTA study group to provide server platforms, operating systems and a set of basic applications. This is a similar pattern of deployment to that implemented in ScholarNet⁵⁸. It was again found that open source products provide a suitable infrastructure and application framework for classroom use. In the support of learning and teaching, most of the programs used were generic packages employed for various purposes – graphics editing (The GIMP), musical composition (Audacity), media playback (IrfanView) and so forth. It is noted, however, that very few of the programs used were content or subject specific⁵⁹. Whilst six of the eight schools using open source products cited total cost of ownership savings as their prime motivation for switching from commercial offerings, the transparency and flexibility of the open source distribution model was also described as a significant benefit. The implications of adopting a particular software product are reduced because open source resources that are found to be unsatisfactory can be easily replaced whereas the adoption of a commercial product may lock an institution into dependency on one particular technology. In general, schools that were more advanced in their uptake of OSS were able to benefit from existing experience of open source deployment at a staff and management level. A successful transition in primary schools also appears to be driven in large part by the availability of technical support from feeder, or cluster group secondary schools⁶⁰.

Open source software was seen by participants in the BECTA study to provide only indirect benefits in curriculum delivery. These resulted primarily from cost savings which allowed money to be spent elsewhere and, in some cases, allowed for the employment of additional information technology support staff. The fact that open source licenses provide for free use and that they enable students and teachers to use a unified group of applications in school, at home and on their own personal computers would, it is suggested, encourage computer-

56 The British Educational Communications and Technology Agency website:
<http://www.becta.org.uk/>

57 The British Educational Communications and Technology Agency. *Open source software in schools: A study of the spectrum of use and related ICT infrastructure costs*. Available from http://www.becta.org.uk/corporate/press_out.cfm?id=4681, May 2005.

58 The British Educational Communications and Technology Agency. *Open source software in schools: A case study report*. Available from http://www.becta.org.uk/corporate/press_out.cfm?id=4681, May, 2005.

59 See iii.

60 See iii.

based learning in the home and would also allow parents to become involved in the modern learning process to a much greater degree than is currently seen.

It seems, then, that a distinction between the success of OSS products in delivering technical platforms and their employment in direct support of learning and teaching is justified. There is clearly growing institutional acceptance of open source software where it provides operating systems, server platforms and productivity applications. This perspective has probably developed as open source products become mature, as they gain publicity and media attention by threatening the dominance of commercial companies⁶¹ and as high-speed Internet access in classrooms becomes the norm⁶². It is also worth noting that financial pressures in national education systems may tend to demand savings in relatively high-cost areas of provision, such as information technology⁶³. However, the availability of products which provide educational experiences, which enrich learning and provide curriculum support, is limited and the idea that OSS can be harnessed as the basis for better products does not seem to have taken hold as yet. An appreciation of the flexibility of OSS resources does not, according to BECTA, correlate with an understanding of the rationale or existence of a 'free-to-use' software development model and large global community which encourages collaborative product creation and tends to foster the release of potentially novel software products that may not be considered economically viable as commercial offerings⁶⁴. The global nature and size of the open source community provides a diversity of projects which answer a huge range of requirements, their development being motivated by personal interest and expertise rather than the imperative for profit and widespread adoption. It is, of course, important to evaluate the quality of individual initiatives and their potential applicability as educational resources, or as elements in educational software solutions. However, our position is that an understanding and ongoing appraisal of the range of OSS projects can lead not only to the creation of useful and relevant educational products but also to a diversification in and specialization of computer-based learning materials which may enrich all types of education.

Although British schools do not yet make extensive use of open source e-learning and educational software resources, efforts are underway to coordinate provision in this area. A number of initiatives attempt to provide coherent sources of information for those interested in applying OSS materials in learning and teaching. SchoolForge⁶⁵ is a portal site which seeks to organize disparate open source projects and support communities into a coherent consortium that can provide software and learning materials suitable for classroom use. The site was established in order to prevent OSS projects from inadvertently competing in this area where collaboration is in fact required. SchoolForge is international in scope and comprises a large number of projects. It also features online interest and discussion groups for individuals involved in education and software development. A brief examination of the software directories available here indicates that a growing number of OSS resources are becoming available. There is still a significant focus on administrative tools and platform-level software but e-learning materials suitable for student use do feature prominently. One problem is that many of the projects mentioned are not intended as bespoke solutions for education. There are, of course, examples of particularly apposite resources, which include a free version of the Logo programming language, a tool for annotation of foreign language texts, an automated dictation system, a flash-card revision tool and a multimedia storybook. Nevertheless, we perceive a distinct lack of learning materials which are specifically intended to support the delivery of a curriculum.

The above is not a criticism of sites like SchoolForge. They provide an extremely valuable resource for educators interested in exploring and deploying open source software. Such directories serve to highlight the fact that many innovative OSS projects offer possibilities as educational tools. The challenge is to harness the diversity of available projects so that they may be integrated into environments and utilized in developments which are specifically designed to answer identified educational priorities and to provide quality assured resources intended primarily for use by schoolchildren and teachers. It is suggested that many open

61 The Economist. *Microsoft at the power point*. Available from http://www.economist.com/business/displayStory.cfm?story_id=2054746, September 11, 2003.

62 The British Educational Communications and Technology Agency. *Broadband in schools – Becta leads on developing network management standards*. Available from http://www.becta.org.uk/corporate/press_out.cfm?id=3139, January 8, 2004.

63 BBC News. *Rise is not enough say teachers*. Available from <http://news.bbc.co.uk/1/hi/education/1778154.stm>, 23rd January 2002.

64 See iv.

65 See the SchoolForge website. <http://www.schoolforge.net/>

source products categorized as e-learning resources are in fact much broader in scope, are not necessarily designed with learners in mind and would require significant information technology expertise and teacher intervention to be successfully applied in a classroom or distance learning scenario. Participants in the BECTA study commented that open source software tends to look 'less finished' or polished than commercial alternatives. This confirms our experience that successful educational software must be tailored precisely to the needs of the target audience and that quality assurance, testing, the creation of suitable supporting resources and a focus on aesthetic appeal are crucial in ensuring the uptake of a particular application.

There is, it seems, something of a tendency to 'reinvent the wheel' in the creation of educational software. Several companies, for example, offer word processors and spreadsheets that are specifically designed to be used by schoolchildren⁶⁶. There is clearly a need for such software despite the fact that the general marketplace is replete with office and productivity applications. Many mass-market products are not sufficiently easy to use, especially for younger users, and they present too many advanced or redundant features which serve to complicate use and necessitate a relatively steep 'learning curve'. The mantra that educational software should be about education and not grappling with a technical system is always of critical importance. It is, of course, easy to underestimate the adaptability of learners, their willingness to adopt new software and capacity, especially as far as schoolchildren are concerned, to quickly come to terms with interfaces which may be considered complex. One finding of recent research⁶⁷ is that educators are drawn to open source software exactly because it offers a broader range of technical experience than is available through reliance on a defined set of applications. Moving to the Linux operating system is attractive in part because it represents a distinct change from the dominant software products and students are often willing to come to terms with the new environment because it broadens their experience and provides a degree of novelty. Nevertheless, it is important to create educational software and computer-based resources which are primarily educational, which prioritize the process of learning in a specific area and which may lead to the adoption of broader or more complex materials as the ability of the learner grows. To this end, it is suggested that the effective integration of open source software into educational products provides an opportunity to focus on the pedagogical aspects of development. Time and resources can be better spent on tailoring a solution to particular educational priorities and learning objectives where a coherent and proven technical platform is already in place. Any competent software developer can produce a word processor for use in schools but a smaller proportion will be able to deliver a product which is genuinely educational, which encourages learning and which supports students and teachers in their developmental objectives.

This is not necessarily an easy balance to strike, however. The hypothesis that freely-available software can simply be tailored to produce effective educational products is true only in certain cases. Our experience shows that there is a plethora of interesting open source projects which may, in theory, contribute to the creation of genuinely novel and interesting curriculum materials. However, the development of such resources is only feasible where existing work has occurred in a structured and coordinated manner. The identification of an established and well-supported open source product on which to base the creation of educational software is critical since adoption of a small or technically poor platform can serve to complicate and hinder further work rather than to facilitate it. The process of identifying educational requirements in a given context and of ensuring that a software application addresses these is also far from trivial. To this end, fostering relationships with individuals in the open source community who are intimately familiar with the technicalities of an established platform must be allied with specialized educational knowledge. The involvement of practitioners, students and educational experts who are prepared to adopt and test a product during the development phase is also required in order to create an effective learning resource. Such a division of the project team allows for an appropriate application of different expertise. Individuals who have technical and domain-specific knowledge are able to concentrate on development and adaptation for the particular purpose at hand, confident that the underlying code is already stable and reliable. Educators are included from an early stage and are able to ensure that the resource is tailored to meet their needs. This coordinated effort should therefore produce a tightly integrated product which, if the development process has been managed effectively, will be appropriate to the target audience.

Some of the considerations mentioned above are self-evident. Perhaps the greatest

66 TextEase from SoftEase Software is an example of this. The SoftEase website is available at <http://www.softease.com/>.

67 See iii.

challenge involved in using open source resources in this context lies in overcoming commercial and institutional reservations about the nature of such software and the possibility that financial and intellectual investment in the adaptation and enhancement of freely-available code will be unsecured. To this end, it is important that procurement bodies, agencies and software companies adopt a creative and flexible approach to development. There must be sufficient confidence in the ultimate quality of a product and its value in educational terms that the intellectual worth of the finished article is seen to lie in the very work undertaken to make it an appropriate curriculum tool. The seeding of enhanced open source software back to the original community then becomes a very positive step. Development of the underlying components will continue and this ongoing process of improvement can easily be harnessed in future versions of the educational package. If the diversity of the open source community is to be fully exploited, then the principle that domain-specific expertise and technical ability is best externalized and left to those who had the vision to begin a project in the first place will need to be accepted.

A return on financial investment and a coherent system for protecting the intellectual property vested in a resource is clearly required. This speaks more about the importance of an effective educational development methodology than it does of a need to fanatically protect the distribution of code flowing from a commercial software project or a need to limit the inclusion of external developers. This is a state of affairs of which companies in the general software marketplace are becoming increasingly aware. Sun and Netscape have both seeded the source code of several applications to the open source community and they actively support the ongoing involvement of community members⁶⁸. It is thus possible to reduce the burden of technical development and the financial expenditure required to release new versions of a major software product. Sun's active support of a large international open source community now ensures, for example, that OpenOffice.org⁶⁹ has come to rival Microsoft Office in terms of features, user experience and flexibility, whilst continual improvements made to the application source code are periodically drawn in to Sun's commercial StarOffice application.

In the following sections of this paper, we consider two recently commissioned software projects that have and are being developed under the National Grid for Learning programme in Scotland. The first of these is OpenOffice.org Gaelic, a fully localized Scottish Gaelic version of the open source productivity suite. The second is PhraseBox, which is still under development and will provide a large-scale corpus search engine for use by pupils and teachers in the third and fourth years of secondary education. OpenOffice.org is an important infrastructure product in the classroom. It is an interesting case study because the existence of an industrial-strength OSS office suite allowed for provision of a vital resource for minority language speakers where issues of resourcing and limited finances would have precluded bespoke development. PhraseBox, on the other hand, involves the application of groundbreaking linguistic research and domain-specific expertise in an educational product which has a potentially large market.

OpenOffice.org Gaelic

OpenOffice.org⁷⁰ is a multiplatform, multilingual office suite that is free to download, use and distribute. It provides an almost complete alternative to the Microsoft Office family of products and is file-format compatible with that software. The English version of OpenOffice.org is already used quite widely in schools. It comprises a word processor (Writer), a presentation manager (Impress), a spreadsheet (Calc) and a drawing package (Draw). The only significant omission is a database module. Microsoft Office Professional includes Access, the industry standard database application and, whilst OpenOffice.org can interface to a wide range of database servers, no standalone capability is currently offered. Sun Microsystems released the source code to their StarOffice product in July 2000⁷¹, less than twelve months after acquiring the German software company that originally created it. Collab.Net was commissioned to create and maintain the community site, which includes development tools such as source versioning, source browsing, development mailing lists and issue tracking.

68 Sun Microsystems. *Sun open sources StarOffice technology*. Available from <http://www.sun.com/smi/Press/sunflash/2000-07/sunflash.20000719.1.html>, July 19, 2000.

69 See the OpenOffice.org website. <http://www.openoffice.org/>

70 See xv.

71 See xiv.

Since the release of StarOffice 6.0, Sun has been building their suite using the OpenOffice.org source code, application programming interfaces, file formats and reference implementation. Coordination of the development process by project managers at Sun ensures continued product evolution and the maintenance of effective relationships between key members of the international development team. Version 2 of OpenOffice.org is currently at the release candidate stage and the final product launch is expected imminently.

In October of 2003, Learning and Teaching Scotland⁷² issued, on behalf of the Scottish Executive Education Department (SEED), a competitive tender through the Digital Content Procurement Programme. This called for the production of a word processor and supporting curricular documentation in Gaelic. The demand for such a product arose from the 2003 annual conference for Scottish Gaelic teachers and educators. Conference identified a strong requirement for the creation of a word processor in Scottish Gaelic as school children learning in Gaelic-medium and Gaelic teaching schools throughout Scotland had no option but to use English language computer software for common productivity tasks. This was seen to dilute the system of immersion language education practiced in many schools. Part of Learning and Teaching Scotland's remit is to provide independent advice on educational technology for ministers in the Scottish Parliament and to encourage the creative and effective use of information technology in education. It was the considered view of the LT Scotland team involved in the Gaelic word processor project that OSS represented a potential means of fulfilling that part of the organization's remit. However the rules governing public sector procurement precluded the overt specification of an OSS solution although strong indications were given that such was the preferred option. The winning bid proposed the localization of OpenOffice 1.1.1. This was provided much greater functionality than had been requested in the Invitation to Tender and represented a significant increase in the investment return both to the public purse and to the Gaelic speaking community in Scotland. In November 2003, Cànan Ltd was awarded a contract by Learning and Teaching Scotland to produce a Scottish Gaelic version of OpenOffice.org.

As a living language, Scottish Gaelic is now largely confined to the North West Highlands of Scotland and the Island communities of the Inner and Outer Hebrides. There are also sizeable communities in the Scottish 'central belt', particularly in Glasgow and Edinburgh. Census data shows that the number of Gaelic speakers over the last century has dropped continuously and dramatically. In 1891, 254,415 Gaelic speakers were recorded, representing 6.84% of the Scottish population. Of these, 43,738 people were monoglot and 72% of speakers lived in the Scottish Highlands. A century later, the statistics had changed dramatically. Only 65,978 speakers remained, representing merely 1.37% of the population, and there were no monolingual speakers left. Some thirty years had passed since a figure for Gaelic-only speakers was recorded - 477 in 1971. The 2001 Census results were awaited with apprehension in Gaelic communities, as it was widely anticipated that the number of speakers would fall to 55,000 or less. In fact, the figures were slightly better than expected, with 58,552 Gaelic speakers recorded, representing 1.21% of the population of Scotland. This 11% drop in the speaker base over the last decade remains significant, however.

By 1946, some schools in Glasgow were introducing Gaelic into the curriculum for the first time, and within twenty years a Scottish Leaving Certificate examination paper in the language was available. The first Gaelic medium units were opened in 1984; one in Inverness and one in Glasgow, and today their number has risen to sixty. Fourteen secondary schools currently present candidates in Gaelic (for Proficient Speakers) at Standard Grade (equivalent to the English GCSE qualification), Higher Grade (equivalent to A-Level), National Certificate and Certificate of Sixth Year Studies levels, while twenty eight present candidates for Gaelic (Learners)⁷³. Taken together, there are approximately 5,500 children in Scotland studying or being taught the language through these various channels. Outside the school system, Sàbhal Mor Ostaig, a further and higher education college based on the Isle of Skye, provides Gaelic-medium instruction, to degree level, in various disciplines.

Creating a Gaelic office suite for use in Scottish schools presented a significant challenge. There is little other software available in the language and certainly nothing of the size or scope of OpenOffice.org. The project thus presented a first opportunity to provide an industrial-strength application for Gaelic speakers and learners. As such, the majority of development time was dedicated to translation of the program interface, which totals about 96,000 words. Most of the technical terms we encountered had no existing Gaelic equivalents

72 Learning and Teaching Scotland website: <http://www.ltscotland.org.uk/>

73 See the Scottish Parliament website.

http://www.scottish.parliament.uk/whats_happening/research/pdf_subj_maps/smda00-10.pdf

and it was therefore necessary to coin new terms which would be both understandable to the target audience and also consistent with any precedent that did exist. A previous localisation of the Opera web browser, completed by staff at Sàbhal Mor Ostaig, served as a limited foundation for certain aspects of the localisation but, again, the language content of a web browser is significantly narrower than that of a full office suite. The decision to use OpenOffice.org as the basis for the Gaelic product was taken primarily because it provided a proven and stable technical platform which offered all the features that would be required at school level. Although the original specification called only for a word processor, high levels of activity evident within the OpenOffice.org community afforded the opportunity to provide valuable additional functionality in the same timeframe as would be required for bespoke development of a single software module.

At the time of project inception, OpenOffice.org was already available in some eighty eight world languages. The international nature of the community and an existing technical framework for localisation meant that procedures and tools for translating the software into other languages were already well defined, supported and documented. OpenOffice.org localisation occurs in partnerships centred around native language community sites that are accessed through the main project portal⁷⁴. Our project team was relatively small, consisting of only three individuals and work was completed within twelve months. We benefited from the experience and support of many other people, however. Assistance was received from as far afield as the Czech Republic, Wales, Ireland and South Africa. The establishment of these external relationships proved critical to the success of the venture as it enabled us to delegate and manage the technical elements of development, whilst maintaining an in-house shadowing capability. A focus on linguistic and pedagogical considerations was thus achieved and this helped to ensure that OpenOffice.org Gaelic effectively fulfils the requirements of the target audience. Issues of program stability and platform compatibility were essentially removed since the fundamental elements of the package are already widely distributed and are subject to ongoing testing procedures.

It has previously been observed that mass-market software offerings are often too broad in scope or too complicated to be applied effectively as educational tools. OpenOffice.org certainly provides an extensive range of features and functionality, much of which would not be used in an educational environment. In order to make the Gaelic product accessible and relevant to our target audience, we prioritised the development of supporting resources and tutorial content. This emphasises and demonstrates elements of the software that would be of most relevance in Gaelic schools. The in-program help content was completely re-written and, in general, shortened. A comprehensive curriculum guide, which exemplifies application of the software in support of the 5-14 National Curriculum for Information Technology⁷⁵, was also produced. Finally, a Getting Started guide and interactive introductory materials were developed in order to ensure that the package was as accessible as possible and that schoolchildren would be quickly able to make use of the new tool. In total, the project involved some 120,000 words of language translation, 30,000 words of which was new content written specifically for young learners.

From the outset, it was clear that linguistic and educational issues would be of greatest importance. We wanted to ensure that the project did not become, and was not considered primarily an exercise in technical development. The decline of Gaelic as a living language means that opportunities to provide modern software tools for learners and speakers will not often arise and our intention was to establish a substantial benchmark for such activities in the future. This would serve to establish a base of Gaelic computing and software terms from which other projects could benefit. At the same time, we recognise an imperative for linguistic consistency in these endeavours. Because much of the phraseology encountered will be new, it was vital to achieve a consensus about language as far as possible. This will stand the language in good stead as future software packages now have a clearly defined and relatively widely accepted standard from which to work. The scope for disagreement and diversity of preference in the expression of even relatively straightforward but linguistically novel phrases was not surprising but it made the task of establishing consensus relatively difficult. The issue was tackled through the establishment of a language group who met regularly during the development process in order to review language points and to resolve disagreement about appropriate terminology. This system worked well and it also resulted in a clear list of priorities for the refinement of the translation and language usage in future versions of the software.

74 See the OpenOffice.org Native Language Confederation's list of supported languages at <http://l10n.openoffice.org/languages.html>

75 See the 5-14 National Curriculum for ICT homepage: <http://www.ltscotland.org.uk/5to14/curricularareas/ict.asp>

The development process culminated in an extensive beta testing programme. OpenOffice.org Gaelic was distributed widely to educators, language experts and interested individuals. They were then able to record feedback about the performance and suitability of the software through an online issue tracking system. A large number of comments were received and it is a testament to the quality and stability of the OpenOffice.org platform that the vast majority of these related to language use. Once again, our decision to use an established suite of programs was vindicated in that testing time could be spent analysing the educational worth and appropriateness of the new tool rather than on discovering and resolving technical problems. The development team addressed all issues as they were submitted and the centralised issue tracker proved to be a catalyst for further debate which necessitated a detailed evaluation of linguistic consistency and accessibility.

The OpenOffice.org Gaelic product was distributed to all Scottish schools in August 2005. It was extensively demonstrated at SETT, the Scottish Learning Festival⁷⁶, which was held at the Scottish Exhibition and Conference Centre⁷⁷ in Glasgow this year. Pupils from Greenfaulds High School in Lanarkshire⁷⁸ were on hand to exhibit work that they had created using the new software and their teachers had been involved in the quality assurance process prior to release. The product has been widely welcomed since its release and is already being used in schools. Despite the success of this product, it is important that the development is not treated as an isolated event. A certain amount of ongoing development will be realised through the open source community, to which our localised content and amended codebase was contributed upon project completion. It is to be hoped that informal efforts through this channel will help to keep the software up-to-date as future OpenOffice.org versions are released. The reality is that ongoing commercial commitments to updating the product will be relatively small if a programme of incremental improvement is established and financed. Initial indications are that members of the OpenOffice community will be eager to continue work on the Gaelic localisation but the scope of this work will become more significant as the main application develops ahead of the Gaelic product.

PhraseBox

PhraseBox is a large-scale corpus search engine designed for use in schools. It is intended to highlight phraseological patterns and effective linguistic structures that pupils can adapt and use in their own work. At present, the system derives information about current English from part of the Bank of English corpus⁷⁹ with additions from Newsquest UK⁸⁰. A PhraseBox user enters one word, two words or more as their initial query. The system returns a distillation of the way the word or words is used, presented in established patterns of phraseology. A more detailed analysis of the way in which the words fit together can be obtained, and many refinements of the original query are available as the user investigates deeper into the structure of the language.

The technical problems which had to be solved relate to the distributional patterns of words in texts. To return a reliable and summary account of the way in which words are used, the computer requires a very large amount of data in order to sift through it, identify the recurrent patterns, and evaluate them relative to the needs of the user. Given a "telephone number" size of corpus, in the hundreds of millions of words, the size of a potential results file is often enormous, and it has to be further analysed and stripped down to make it usable in the classroom, without losing the breadth or diversity of patterning. All this has to be done in real time, and the very strong patterns of the very frequent grammatical words have to be restrained to allow the less frequent but equally important patterns of vocabulary combinations to come through. PhraseBox allows users to search either the corpus as a whole or to search on specific genres within that corpus. Thus it is possible to tailor results to a particular type of writing – sports reporting, tabloid news reports, broadsheet coverage and so on. This provides a very effective account of language use in different media and for different audiences. The initial objective here is to provide a resource that both supports the teaching of English and also helps to improve pupil literacy. We are currently developing a prototype of the platform which will be released to Scottish schools next year. This project employs open

76 The SETT homepage. <http://www.ltsotland.org.uk/sett/>

77 Scottish Exhibition and Conference Centre homepage. <http://www.secc.co.uk/>

78 Greenfaulds High School homepage.
<http://www.northlan.gov.uk/education+and+learning/schools/secondary/greenfaulds+high.html>

79 See the HarperCollins Bank of English website.

<http://www.collins.co.uk/books.aspx?group=153>

80 See the Newsquest UK website. <http://www.newsquest.co.uk/>

source software to facilitate the indexing and searching of large volumes of text. PhraseBox otherwise represents the application of novel research in the fields of corpus linguistics and natural language processing and it is the first significant attempt to harness such technologies for mainstream education. The system has been designed to be corpus independent which means that the text sources used can be easily augmented and changed. PhraseBox will therefore diversify, possibly to include different languages and a wider range of different types of writing than there is at present in the corpus, once the initial concept is proven and it becomes accepted as a useful resource in schools.

Our development methodology for this project recognises the importance of preserving intellectual property in the novel aspects of the system. Once again, Cànan has been commissioned to create the service by Learning and Teaching Scotland and we are working with the LTS consultant, Professor John Sinclair⁸¹. Our substantive work in this area will create a product that handles complex user queries about language use through a simple and accessible user interface. Results will then be derived through the application of statistical methods that provide concise and relevant information about the most common forms of language use and effective grammatical constructions. Such a separation of functionality means that, as the underlying libraries are developed and enhanced within the open source community, they can be easily re-integrated with PhraseBox and new features will be harnessed for use in schools. The expertise of researchers in corpus linguistics and natural language processing is thus made available and presented as a practical tool that assists in the development of literacy and in the support of language teaching.

PhraseBox presents a method of using open source software in education which differs from the norm. It is a tool for curriculum support which successfully employs highly complex and domain specific software by embedding it in a service that is designed to be used by teachers and schoolchildren. This is not a resource for linguists that can be used to educate by those with knowledge of the underlying disciplines or statistical methods. PhraseBox has been specifically created for use in classroom teaching or in a distance learning scenario. Access to a large scale corpus will provide pupils with demonstrations of natural language use rather than stilted examples. They will be exposed to a diversity of linguistic expression and a variety of writing styles which, it is hoped, will enable them to appreciate both the power and subtleties of language. The ultimate objective is to create an open ended reference resource which encourages linguistic experimentation and fosters richness in original work.

A successful implementation of the PhraseBox project will, it is suggested, prove the concept that novel technologies can form the basis for useful and pedagogically valid educational software. The process of forming a development team with the requisite skills and contacts to first understand and then encapsulate such materials is difficult and may not occur frequently. However, it seems evident that the open source community provides a wealth of research-driven tools and resources which may theoretically be exploited in this manner. It is again true to say that the success of such an approach depends upon the confidence and vision of procurement bodies and software companies in identifying, financing and supporting the development of apparently marginal or specialised OSS tools.

Summary

Open source software is gaining acceptance and becoming more widely deployed in education. Financial pressures and a desire to diversify student experience will, it is suggested, continue to drive a move away from commercial products, particularly in the provision of operating systems and generic application programs. As more educationalists and teachers start to use OSS resources, so the levels of support available for practitioners will increase and this will further encourage their use. The requisite infrastructure for high-quality and responsive technical support already exists courtesy of systems currently employed to foster collaboration within the open source community, albeit that the methods of issue resolution may be somewhat less accessible or 'polished' than those offered by commercial companies. One attractive possibility offered by the OSS movement is for the creation of substantive resources that support minority education. OpenOffice.org Gaelic is an important productivity application suite for students and teachers of Scottish Gaelic. This has been released in a situation where a lack of resources, a small potential marketplace and a paucity of technical development expertise would otherwise have prevented the bespoke creation of an equivalent resource. In commercial terms, using open source software for this project was a positive factor. It allowed the immediate development team to focus on educational and linguistic issues whilst substantially devolving technical adaptation and compilation responsibilities to external individuals who were intimately familiar with the processes

81 See the Tuscan Word Centre homepage. <http://www.twc.it/>

involved. Such an approach places an onus on the effective management of the development process and on an ability to shadow the activities of external consultants for reasons of commercial security. However, the success of the OpenOffice.org project speaks of a much broader potential to support minority language education with modern resources to an extent not previously possible. The ongoing challenge here is to ensure that the development of such resources continues and it is suggested that this can be achieved in partnership between the open source community and national educational procurement bodies.

PhraseBox will harness academic research and specialist software tools in an e-learning resource for curriculum support and linguistic development. We perceive an important distinction between the employment of OSS resources in the provision of a technical infrastructure or generic computing platform and their use to enrich and improve educational experience. Once again, the creation of appropriate supporting resources and a focus on specific educational requirements will be a key to the success of this service. However, the principle that partnerships between subject experts, software developers and educational specialists can serve to expand and redefine the provision of e-learning in certain areas is an interesting and apparently valid consideration. Both projects highlight the potential benefits flowing from an abstraction of fundamental technical delivery mechanisms from the contextualizing work of creating an educational resource. The intellectual and financial commitment required to create effective e-learning software means that methodologies which enable a clear focus on issues of pedagogy and subject relevance can result in the development of better resources. This is not to say that existing educational software is of poor quality. Rather we perceive substantial opportunities to exploit research and specialist tools in areas that may be considered too marginal, specialized or complex for integration into traditional mass-market educational products.

In conclusion, then, it seems logical to propose the fostering of comprehensive open source frameworks for education. This would build upon and draw together the growing number of OSS resources which are relevant to learning and teaching in different areas, combining existing operating systems, productivity applications and curriculum resources with more specialized and subject specific tools designed to develop and enhance knowledge, to support teaching and to provide freely available and flexibly licensed computer-based e-learning resources. It is to be hoped that continued procurement of educational products based upon open source software will gradually result in a more comprehensive choice of applications and materials that are focused on the different strands within mainstream education. A properly managed national or regional open source development policy should facilitate the creation of effective and innovative e-learning and educational software products which offer best value and which remain contemporary and relevant because they are freely adaptable.

Hamlet goes XML: CrossAnnotationLinking and Personal Learning Experiences

Benjamin Birkenhake¹, Stefanie Panke², Andres Witt¹
Universität Bielefeld¹, Institut für Wissensmedien (IWM)²
benjamin.birkenhake@gmail.com, s.panke@iwm-kmrc.de, andreas.witt@uni-bielefeld.de

Abstract

XML-based technologies offer powerful resources for open source applications in the field of e-learning. The paper describes a model of hypertext as interlinked structures that can be intertwined by cross-annotation linking. This infrastructure integrates multiple perspectives and allows creating a personal learning environment. We exemplify the approach in a case study: the Hamlet project. In the course of this project, several German translations of William Shakespeare's Hamlet have been collected and annotated. Two different annotation layers are used to achieve a cross-linking reference between the various German translations. We will describe the theoretical background of cross-annotation linking and the actual technological implementation of the system. Additionally, we will use the personas method to gain insights into the potential benefit of the system as a personal learning environment.

1. Introduction

Although in theory open source and e-learning mix quite happily, the day-to-day practice often presents a different picture. Users have a more or less justified prejudice about open source being "somewhat difficult" and only for "techies". In this article we will reflect upon typical problems and propose a treatment for curing. We will describe the development of an XML-based environment from two different perspectives: The technological point of view and the user-centred design perspective.

XML allows combining single source storage of data with multiple modes of presentation. This offers rich opportunities for e-learning with respect to personalization and information retrieval. Since locating information is a vital part of the reading process (Guthrie & Mosenthal, 1987), functions that allow for the retrieval and management of resources are core features of an e-learning environment. To reveal the full potential of XML-technology for these purposes, we had to shift our traditional view on hypertext as interlinked segments of text to a new metaphor of interlinked trees which is exemplified on the basis of the case study Hamlet (see section 2).

The digital technologies – especially mark-up languages – have changed the way information is managed and a growing body of literature has become available which tackles the technical, conceptual and pragmatic consequences of this development. Information retrieval focuses on structuring and presenting information in the digital environment and covers different aspects of organizing information like classification and taxonomy, indexing and thesaurus construction, as well as metadata (Marchionini, 2004). Information seeking models describe the processes of finding information to fill a certain knowledge gap from a user's point of view (see Marchionini, 1995; Ellis & Haugan, 1997). A wider perspective on information activities, including publishing and knowledge exchange, is given by theories on information behaviour (see e.g. Hektor, 2003). To reflect upon the information behaviour of the potential users of the hamlet corpus, we will deploy the personas method and match fictional users' needs with actual and desirable features of the Hamlet environment.

2. Hypertext as Interlinked Structures: The Case Study Hamlet

The way we search the web or other digital environments is contingent to the explicit or implicit concept of hypertext, which designers have in mind while developing digital artefacts.

Since more and more software is available to conveniently produce hypertext environments, hypertext has become a widely-used (text-)technology. At this point, new metaphors are needed for innovative features, especially for applications in the field of data mining and content management. In the following we will outline a perspective on hypertext as multi-rooted trees which are intertwined by cross-annotation linking (see Witt, 2005a; Witt 2005b).

2.1. Multi-Rooted Trees and Cross Annotation Linking

What is cross-annotation linking? This question is answered best within a historical flashback: The discussion of the first OHCO (Ordered Hierarchy of Content Objects) model (see De Rose et al., 1990; Renear et al., 1996) has shown that textual data could not be understood as a merely single hierarchy of content objects. This evidence was underlined by the emergence of new technologies. The dissemination of SGML and XML put forth an increasing application of multiple annotations of one source of textual data. These multiple annotations usually contain different layers of information, e.g. textual structure and linguistic, philological or narrative information. The different annotations form several trees above the primary textual data. The branches and leaves of these trees can be related to one another in several ways: One or more branches of one tree can be part of a branch of another tree. Two branches can have the same starting point and a different ending, as well as different starting positions and an identical ending - which means that branches can overlap (see Durand, 1999; Duruseau & O'Donnell, 2002). This typology of relations - together with the schemas of the different annotation layers - can be used to (semi-)automatically generate hyperlinks between the nodes of the different trees. This allows creating a network of multiple perspectives on one text being linked to one another. As a result, hypertext is no longer based on links between nodes, but offers a reference mechanism between perspectives.

2.2. The Case Study Hamlet

Although Shakespeare's Hamlet is obviously not a unique Hypertext, it is an interesting object to test cross-annotation-linking and several other hypertext-technologies. There is no original edition of Hamlet, which was authorized by Shakespeare during his lifetime. We only have different print editions, which all have a different status concerning their quality, overall length, content and storyline. The most important among these are the so called first folio, the first Quattro and the second Quattro edition of Hamlet. During the centuries editors tried to combine these early Editions to the best Edition possible. A comparable situation exists within the field of German translations of the play. Almost every Translator used several of the early English editions as a basis for a new translation. This leads to a situation in which almost every German or English edition of Shakespeare's Hamlet is a composition of several sources. The relation the editions have with their sources and with each other form a wide network, which could be presented in an e-learning-environment.

Another interesting aspect of Shakespeare's Hamlet is the outstanding position the play has within the western culture for centuries. Hamlet is the single most researched piece of literature, has been analyzed in from various perspectives and is a part of western education. This leads to the request, that a digital environment should represent the variety of perspectives on the play. As part of a PhD thesis, the most important English editions and several German translations of William Shakespeare's Hamlet have been collected and annotated in different ways. This leads to a corpus of Hamlet editions in which each text may exist in multiple forms.

Basis for the XML-annotations are text files, which are transformed to XML using regular expressions. The basic XML-format is TEI 4 drama base tag set. TEI 4 is a major open source concept of the Text Encoding Initiative. It contains a modular Schema for several document-classes. The drama base tag set offers almost all tags needed for a general, formal annotation of a play. In order to provide an easy to annotate mechanism to represent the translation- or origin-relation between lines or paragraph within editions on the one hand and the sources on the other hand, some attributes were added to TEI by us.

The TEI-annotated documents are used for further annotations and presentation. The TEI-documents were automatically enriched with further mark-up, using an open source "auto-tagger". This auto-tagger annotates single words, including the part of speech and the principle form. The TEI-documents are also the basis for the XHTML-presentation. As the TEI-structure contains all information necessary for a graphical presentation, these documents are transformed to XHTML, which is used to present the corpus. This transformation is made with

several XSLT-Style sheets. In the same way XSL-FO is used to generate PDF-versions of each edition. Table 1 illustrates which annotations are provided for the different editions.

Edition	Txt	TEI	XHTML	STTS	Narration
1st Folio	✓	✓	✓		
1st Quattro	✓	✓	✓		
2nd Quattro	✓	✓	✓		
Moby	✓	✓	✓		
Wieland	✓	✓	✓		✓
Schlegel	✓	✓	✓	✓	✓
Fontane	✓	✓	✓	✓	✓
Hauptmann	✓	✓	✓	✓	

Table 1: annotation layers for the different editions.

In many cases translators have re-arranged the flow of stanzas or the course of action. Therefore it is useful to provide an alternative linking mechanism, which does not only focus on the language and the formal structure, but also on the plot. To provide this reference the narrative information is annotated in another layer. This allows to find the same event in different translations of the play. The narrative annotation layer basically consists of events, which can be seen as the smallest elements of the plot. Several events form an action. These can be grouped into more complex actions. All elements are embraced by the root element translation.

Obviously, events may start within one line and end several lines or even speeches later. Since the narrative structure is overlapping with the TEI, both could be stored in separate annotations. Scenes can provide a meaningful unit for basic parts of the plot. Thus the formal and the narrative annotation are semantically aligned - in addition to their reference on identical textual data. This relation can be exploited by creating links between the concept of a scene and the concept of specific actions. The respective linking mechanism is located on a Meta level: it operates on the schemas themselves and not on their instances. The references are generated mechanically on the Meta level, linking different perspectives together. Readers can explore the relations between events and scenes. The procedure could also be used to create a recommendation system as e.g. proposed by Macedo et al. (2003): the annotation integrates the knowledge of experts on narrative structures in the play Hamlet and provides this information to the reader. Figure one shows an example for two different annotation layers which are overlapping.

```

<event>
  <event unit="u0024" sequel="u0030" >HORATIO: Das kann ich; wenigstens das Gerede geht so.</event>
  <event unit="u0013" sequel="u0015">Unser letzter König, dessen Bild soeben vor uns erschien,</event>
  <event unit="u0025" sequel="u0026">ward-wie ihr wißt -vom Norweg Fortinbras - durch wett-
  eifernden Stolz dazu gespornt - zum Kampf gefordert, </event><event unit="u0026" sequel="u0027">in
  welchem unser tapfrer Hamlet (nach unsrem Wissen
  schätzte ihn die Welt von dieser Seite) den Fortinbras
  schlug; </event><event unit="u0025" sequel="u0026"> welcher - nach einem festgestellten Vertrage - [...] </event>
</event>

<sp n="48" f1nr="" q1nr="" q2nr="" moby="48" who="">
  <speaker>HORATIO</speaker>
  <l n="91" f1nr="" q1nr="" q2nr="" moby=""> Das kann ich; wenigstens das Gerede geht so.</l>
  <l n="92" f1nr="" q1nr="" q2nr="" moby="">Unser letzter König, dessen Bild soeben vor uns erschien,</l>
  <l n="93" f1nr="" q1nr="" q2nr="" moby="">ward-wie ihr wißt -vom Norweg Fortinbras - durch wett-</l>
  <l n="94" f1nr="" q1nr="" q2nr="" moby="">eifernden Stolz dazu gespornt - zum Kampf gefordert, in</l>
  <l n="95" f1nr="" q1nr="" q2nr="" moby="">welchem unser tapfrer Hamlet (nach unsrem Wissen</l>
  <l n="96" f1nr="" q1nr="" q2nr="" moby="">schätzte ihn die Welt von dieser Seite) den Fortinbras</l>
  <l n="97" f1nr="" q1nr="" q2nr="" moby="">schlug; welcher - nach einem festgestellten Vertrage -</l>
  [...]
</sp>

```

Figure 1: A multi rooted tree above a single textual data

2.3. Information Retrieval within the Hamlet Environment

As a first result of the multiple annotations, we got a corpus which is based on XML-technology and available via the web. As a second result we developed methods to cope with multiple annotated documents – a task which will probably be necessary more frequently with the growing popularity of XML-technologies. Especially the integration of the narration-annotation-layer can be seen as an example for further parallel annotations. The methods described above lead to an environment which offers different types of users different perspectives on a single, textual object. The following options result from the annotation:

1. The common TEI-annotation allows a structural linking-mechanism between the editions. This allows a user to jump from the first scene in the second act of one edition to the same scene in another edition.
2. Alternatively this annotation can be used to present the user a part of the play in on or more editions of his choice. For example he could choose the second scene of the second act and the editions »moby«, »1st folio«, »schlegel« and »fontante«. So different pieces of text would be presented as parallel columns. This interactive environment is created with PHP5 using the new "fast and simple" XML Extension.
3. The narrative annotation-layer allows several ways to explore a single text or compare some texts with each other. In the first case, the annotation of events and actions provides a way of comparing different editions esp. translations. It allows the user to jump for one edition to the same event in another edition, no matter in which part of the play the event is placed or of which words it is made of.
4. Using SVG an XML-based format for graphics the narrative structure of each translation could be visualized, ignoring the textual basis. This gives an »overview« of plot of the current edition.
5. The introduced concept of cross annotation linking allows us to offer the user automatically generated links from one annotation to another.

With this set of different linking-concepts we present users with almost complete freedom to explore the corpus in a way that fits to their needs. Every layer of information offers a way to access information of another layer in a different perspective. The resulting hypertext structure can be describes as a rhizome. This organization type is characterized by the multi-selective options resulting from a decentralized network of nodes and links. Everything can be linked with everything. In the epistemological philosophy of Gilles Deleuzes and Félix Guattari the rhizom forms the counterpart of the metaphor of the knowledge tree. The latter symbolizes formalized, structured, subject-oriented thinking, whereas the rhizom is a symbol of nomadic thinking (Berressem, 2000).

3. Designing for Learning: The Personas Method

We have seen that cross-annotation linking can provide an infrastructure for personalized learning environments. Nevertheless, the application of high-end technologies can result in very poor learning settings, because the development process is focussed less on the users and their goals and more on the computational challenges of an ongoing project. How can this narrow perspective be avoided or ameliorated? This section describes the personas approach as a method of user-centred design. Personas are fictional users who have individual goals and needs which are reflected in their usage of an environment (Cooper, 1999). They avoid designing for oneself and help the developer to decide which functions will be fruitful for the persons who have to work or learn with the technological artefacts (s) he produces.

The personas technique is an established method in product design (Sinha 2003) and “a powerful complement to other usability methods” (Pruitt & Grudin 2002). The application of personas - fictional people – to represent an abstract consumer has a long tradition in the field of marketing (Pruitt & Grudin 2002). At the end of the past century the use of personas was also heralded in product design (Sinha 2003). Meanwhile the method is applied in different contexts, for example in technical writing to create user documentation (Calde 2004) and is used as well as a design method in software engineering.

What is the advantage of using personas as compared to identifying target groups and designing for these different user groups? The personas approach tries to engage and immerse the designer in the everyday-life of potential users. “Personas help define the product by replacing the abstract, elastic user with the vibrant presence of a specific user who becomes part of the design process” (Sinha, 2003). As a kind of projection foil, personas as elaborated archetypes serve to identify the (information) needs and possible behavioral patterns of the potential users. “Understanding user information needs and mental models is important for design in information-rich domains” (Sinha 2003). Functionalities may be derived easier following the personas needs, interests and possible actions than in abstract design processes.

3.1. Personas for the Hamlet Environment

In order to gain reliable results for the design process, personas cannot be built from scratch – they have to be generated based on data. At this, qualitative data is suitable for the modelling of the personas’ characters and quantitative data may be used – if necessary – for the weighting of the different personas developed (see Arnold et al., 2005). Since in the case study no explicit target group analyses were accomplished, the personas were created on the basis of experience reports: In the context of the Hamlet project several seminars were held within the period from winter semester 2003 to summer semester 2004. The courses on “Cyberhamlet” (<http://www.cyberhamlet.net/>) were concerned with the structure of the annotated corpus as well as a graphic conversion of the drama into a comic. Participants were bachelor and masters students from a wide range of subjects, such as media design, text technology, literature, English Studies, etc. The lecturer was interviewed with respect to his experiences from the courses. According to the objectives and interests of the students who attended these seminars two personas were created. The goal was to model two rather disparate user-profiles, to see how the Hamlet environment would fit to the respective needs. In the following you will meet Beatrice and Gerd, two fictional users with distinctive backgrounds and expectations concerning the interface options provided by the Hamlet environment.

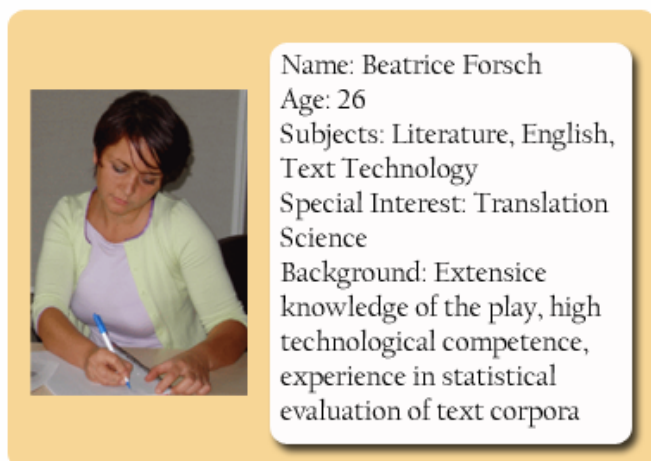


Fig. 2: Persona Beatrice Forsch

Beatrice Forsch, 26, studies comparative literature science, English and text technology at the University of Bielefeld with the goal to obtain a master degree. Her special interest is the translation science. She has advanced theoretical knowledge of English history and literature and has experiences in using computers for statistical analysis of textual data. In her master thesis she wants to analyze and show that each translation is at the same time an interpretation. The material offered within the Hamlet corpus is predestined as a case study, since she can work on the corpus with different statistical methods. Beatrice is especially interested in the scientific expertise coded into the narrative layer of annotation. She wants to compare the flow of events between translations. For further analysis on the differing translations, she would like to conduct semantic field analysis, which could be provided through additional text mining tools. For her purposes it is important that she can export the results of her work in different formats: She wants to print out graphic overviews of the sequence of events to get a quick overview on interesting differences between the translations. To include the material in her thesis she wants to export segments of the graphical maps as GIF, PNG and SVG files. Furthermore quotes from the corpora should be exportable as text files and contain the respective source information.



Fig.3: Persona Gerd Ravig

Gerd Ravig, 23, is a bachelor student of media design at the University Of Applied Science of Bielefeld. Gerd is very practice-oriented: From his point of view, scientific theories should be directly convertible into his creative working context. Together with fellow students Gerd produces filmlets, animation and trick films. He is interested in the material "Hamlet", since he has seen the movie "The Denmark Company", a modern adaptation of the play. Gerd works with the corpus in the context of a project seminar. As a project assignment, he works on a

Comic version of the Hamlet. In the context of the course Gerd would like to complement his practical knowledge in the production of films with theoretical know-how on the design of film script. On the basis the Hamlet environment he is to learn fundamental narrative concepts such as events, actions, chains of events and character constellations. Since Gerd likes to work with graphical representations he uses the possibility to visualize for example event segments from the data. To get familiar with narrative theories, Gerd needs additional didactical tools, e.g. a glossary should clarify the theoretical terms and additional information should be provided which helps him to re-contextualize the action of the play within its historical setting.

3.2. Information Seeking Behavior of the Personas

To identify useful features for an e-learning environment based on multi-rooted trees we apply an adapted version of Ellis' model of information seeking (Ellis & Haugan, 1997; Choo et al., 2000) to specify user tasks within the environment.

Process	Beatrice (Using the Hamlet Corpus as a Tool for Research and Explorative Learning)	Gerd (Using the Hamlet Corpus as an Instructional Learning Environment)
Starting	Refining the research questions she wants to analyze within the Hamlet environment, getting a first overview of the material by retrieving general statistical information on the translations (e.g. number of words, scenes, entries, etc.).	Getting an overview of the play's structure by using the graphical representation as an overview or index. Identifying personal points of interest through mixed modality representations, e.g. a movie version of Hamlet which is linked to the textual data.
Chaining	Following links between the various layers of annotation, identifying points of interest, using the visual representation of narrative structures to compare e.g. Schlegel's and Fontane's translation of the play	Following links between the various layers of annotation, especially exploring didactical context-information and glossary explanations.
Browsing	Investigating specific characters more closely, e.g. by retrieving a selection of all scenes of Rosenkranz and Gldenstern.	Following a certain chain of events, working through a pre-defined learning path which explains and illustrates the relationship between Hamlet and Ophelia.
Differentiating	Identifying and bookmarking specific segments, saving them in the personal profile. Using the data mining filter to scan for the occurrence of semantic fields.	Reading specific segments, express questions and ideas through annotations, saving reading paths.
Monitoring (Adding)	Reviewing previous results stored in the personal profile. Obtaining information about new features or the work of fellow students, learning how to program own queries in XSLT through embedded tutorials, providing own results within the environment.	Re-reading saved paths and personal comments stored in the personal profile. Obtaining information about new features or the work of fellow students, adding own graphical work, using export function to add speeches into the comic produced in the project assignment.
Extracting	Exporting results of analysis through screenshots, quotes and different export formats offered.	Selecting useful pages and sites by bookmarking, printing, copying and pasting, etc.

Table 2: Information Seeking Behaviour (adopted from Choo et al. 2000, 7) applied to E-Learning Tasks within the Hamlet Environment

3.3. Recommendations for Further Development

The Hamlet corpus is still work in progress. A very basic interface has been provided based on the XHTML presentation generated from the annotated corpus. So at a very early stage of development useful features for learning purposes are discussed. The following summary

comprises central recommendations for employing the Hamlet corpus in an e-learning context.

As we have learned from the scenarios of use sketched with the personas, personal interests, learning goals and information seeking motives may differ widely among the potential users of the system. Therefore it is of vital importance that the students can select which perspective on the corpus they want to pursue. They should be able to reduce the complexity of the interface according to their specific needs. Specific browsers should allow to select specific translations and perspectives. Browsing "perspectives" would allow students to switch between e.g. the TEI annotation layer and the narrative information. Browsing "translations" would provide the possibility to compare e.g. the narrative structure of Schlegel's and Fontane's translation. Moreover, visually coded links and a colour-navigation may help to reduce cognitive load and at the same time provide explorative flexibility.

Users like Gerd, who are not thoroughly familiar with the play, will actually want to read parts of the play within the learning environment. Therefore it is important that the web-interface provides reader-friendly typeset, as well as printable versions. To resume the reading process at a later date, students should be able to set bookmarks and annotate single nodes or segments. Furthermore, novice students would profit from pre-defined learning and reading paths, which illustrate a certain question comprehensively, e.g. the relationship between Hamlet and Ophelia. Additional information like glossary items explaining narrative concepts as well as information on the social and historical background of the play would support and deepen the understanding.

Advanced students like Beatrice will use the environment to investigate very specific questions. Hence, filters should be provided to extract meta-information and to assemble and display only specific segments of the textual data, e.g. all scenes of Rosenkranz and Gldenstern. A "data mining tool" could offer the possibility to define semantic fields and analyse different translations of the play accordingly. To give advanced users even more flexibility, tutorials could be produced which explain how to form a query with programming language XSLT, working directly on the annotated data.

4. Conclusions and Perspectives

We have presented a twofold strategy for the application of open source in the field of e-learning: We have described an XML application which reveals opportunities for information retrieval and personalization features. We assume that with growing possibilities to create multi-rooted trees on single-source primary data, the importance of cross-annotation linking will increase.

To provide recommendations for the use of cross annotation linking as an e-learning tool, we exerted the personas approach. Narrative design can serve as an instrument for the development of open-source based and user-friendly e-learning environments. The personas reflect the experience of working with students of the study course text technology at the University of Bielefeld. Further research should lead to a refined picture of the personas generated in this context. In this respect, involving other subjects through interviews with teachers and students may provide an interesting perspective.

To meet the recommendations derived from the personas approach still remains a challenge. Nevertheless, the evaluation of the features based on the personas helped the developers to understand the potential benefits and obstacles in deploying the Hamlet corpus as a self-directed learning environment. It is well-known that interface design is more than "lipstick on a bulldog". However, many technologically innovative projects suffer from weaknesses in providing an aesthetically attractive surface, which offers functionality really needed by its users.

In the case of the Hamlet environment, involving the future user in the design process took place before extensive work on the interface was accomplished. Therefore it is possible to balance between technological expenditures of implementation and didactical desirability. The proposed strategy will result in a feature specification which comprises technological, organizational and user-oriented perspectives.

5. References

- Arnold, P., Geiser, B., Panke, S. (2005). Personas im Designprozess einer E-Teaching Community. In: J. M. Haake, U. Lucke & D. Tavangarian (eds.). *Proceedings DELFI 2005*, 3. *Deutsche e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)*.
- Berressem H. (2000). Unterwegs im Docuversum. Zur Topologie des Hypertext.. In: M. Klepper, R. Mayer & E.P. Schneck. (eds.). *Hyperkultur. Zur Fiktion des Computerzeitalters*. Berlin: de Gruyter.
- Calde, Steve (2004) Using Personas to Create User Documentation. [Online] Available: http://www.cooper.com/content/insights/newsletters/2004_issue04/Using_personas_to_create_user_docs.asp [11/11/2004]
- Choo, C.W., Detlor, B. and Turnbull, D. (2000). Information Seeking on the Web: An Integrated Model of Browsing and Searching, *First Monday*, 5/2, Online: http://firstmonday.org/issues/issue5_2/choo/index.html [Available, last checked 5/9/2005].
- DeRose, S., Durand, D., Mylonas, E. and Renear, A. (1990). What is Text, Really? *Journal of Computing in Higher Education*, 1, 2, pp. 3-26.
- Durand, D. G. (1999). *Palimpsest: Change-Oriented Concurrency Control for the Support of Collaborative Applications*, PhD Thesis, Boston University.
- Duruseau, P. and O'Donnell, B. M. (2002). Concurrent Markup for XML documents. In: *Proceedings of XML Europe 2002 Conference Proceedings* (Barcelona, Spain, May 20-23, 2002). Online: http://www.idealliance.org/papers/xml02/dx_xml02/papers/03-03-07/03-03-07.html [Available, last checked 5/9/2005].
- Ellis, D. and Haugan, M. (1997). Modeling the Information Seeking Patterns of Engineers and Research Scientists in an Industrial Environment. *Journal of Documentation*, 53, pp. 384-403.
- Grudin, J. and Pruitt, J. (2002). Personas, Participatory Design and Product Development: An Infrastructure for Engagement. *Proc. PDC 2002*. CPSR, pp. 144-161.
- Guthrie, J. T. and Mosenthal, P. (1987). Literacy as Multidimensional: Locating Information and Reading Comprehension, *Educational Psychologist*, 22, pp. 279-297.
- Hektor, A. (2003). Information activities on the Internet in everyday life. In: *New Review of Information Behaviour Research* 4/1 (2003), pp. 127 – 138.
- Pruitt, J. & Grudin, J. (2003) Personas: Practice and Theory. In *Proceedings of the 2003 conference on Designing for user experiences*. San Francisco: ACM Press.
- Macedo, A. A., Truong, K.N. & Camacho-Guerrero, J. A. (2003). Automatically Sharing Web Experiences through a Hyperdocument Recommender System. In: *Proceedings of the 14th conference on Hypertext and Hypermedia (Hypertext03)*, (Nottingham, UK, August, 26-30, 2003). Online: <http://www.ht03.org/papers/pdfs/6.pdf> [Available, Last checked: 6/16/2005]
- Marchionini, G. M. (1995). *Information Seeking in Electronic Environments*. Cambridge: Cambridge University Press.
- Marchionini, G. M. (2004). From Information Retrieval to Information Interaction, *Keynote at European Conference on Information Retrieval*, (2004, April). Online: <http://ils.unc.edu/%7Emarch/ECIR.pdf> [Available, last checked 5/9/2005].
- Martzoukou, K. (2005). A review of Web information seeking research: considerations of method and foci of interest. *Information Research*, 10, Online: <http://informationr.net/ir/10-2/paper215.html> [Available, last checked 5/9/2005].
- MIT (w.Y.): *The complete work of William Shakespeare*. Online: <http://www-tech.mit.edu/Shakespeare/works.html> [Available, last checked 5/9/2005].
- Sinha (2003) Persona Development for Information-rich Domains. In: *CHI 2003 Conference on Human Factors in Computing Systems*. Fort Lauderdale, Florida.
- Renear, A., Mylonas, E. & Durand, D. (1996). Refining Our Notion of What Text Really Is: The Problem of overlapping Hierarchies. In: N. Ide & S. Hockey (eds.): *Research in Humanities Computing*. Oxford University Press, Oxford, pp. 263-280.
- Witt, A. (2005). Multiple Hierarchies: New Aspects of an Old Solution. In: Dipper, S., Götze, M., & Stede, M. (Eds.), *Interdisciplinary Studies on Information Structure (ISIS)*, Potsdam: Universitätsverlag Potsdam, pp. 55-86.
- Witt, A., Goecke, D., Sasaki, F. & Lungen, H. (2005). Unification of XML Documents with Concurrent Markup. In: *Literary & Linguistic Computing*, 20 (1), Oxford: Oxford University Press, pp. 103-116.

Implementing and Adjusting Open Source Solutions: the Internet Seminar “Introduction to Applied Computational Linguistics”

Karin Naumann, Lothar Lemnitzer
Seminar für Sprachwissenschaft
{knaumann, lothar}@sfs.uni-tuebingen.de

Abstract

In our paper, we will describe an eLearning scenario that was designed according to the needs of our learning contents and target group, i.e. advanced students and graduates. It is a synchronous setting for discourse-based learning in small working groups. The seminar was centred around an Open Source chat tool called TULKA. For asynchronous cooperation and communication we used the Open Source environment MOST. We will explain how the Open Source Software was implemented into the learning environment and how it was adjusted to our specific needs. The learning material for this seminar consisted of modular learning units. Some of these units entered the virtual course “Computational Lexicography”. We will show, how the Open Content units were implemented into the Open Source environment ILIAS, used in the second course.

1. Introduction

eLearning, having gained maturity as a way of organizing the teaching and learning process, still poses some challenges to learners as well as to teachers. While learners, in the ideal case, gain some freedom to organize their learning process, teachers are confronted with new didactic scenarios which might be looked at as a promise or as a threat. We have always been looking at the promises which eLearning has to make for our teaching activities at a university.

The major challenge we had to master in our practice as eTeachers has been to harmonize the needs and learning styles of our target groups, the learning contents, and the learning context or learning environment. The period in which we gathered the experience that we want to present to you, i.e. the time between 1999 and 2003, has seen a lot of eLearning tools appear and quite a few of them disappear. We are still in an exciting process where eLearning tools do not appear as monolithic 'take-it-or-leave-it' tools, as nowadays is the case with text processing software, but as conceptualizations of the eLearning process which are subject to an on-going dialogue between the developers and the users. This is the more the case with open source software, with which the developers rely not only on the feedback, but also the participation of the users.

The target group of our courses is specific: we are teaching computational linguistics to advanced undergraduate and graduate students. The focus in learning is therefore not so much on the conveyance of factual knowledge, but on the discussion of controversial topics and on the implementation of ideas, formalisms etc. On the other hand, these students form a group with a high technical competence, but with a lack of media competence in computer-based communication and the use of a multimedia learning environment. Our learning environment has to be adapted to this target group. To reach this goal, we had to influence the development of eLearning tools we needed. We have done this in close co-operation with the developers. This was, however, not always a story of success.

With this paper, we want to contribute to this dialogue by presenting our experience with using and enhancing eLearning tools.

2. Organizational framework

The Department of Linguistics at the University of Tübingen has developed two virtual online seminars where primarily synchronous CMC (Computer Mediated Communication) is used for communication and cooperation between students from different German universities. The first seminar is an Introduction to "Applied Computational Linguistics" (ACL) for graduate students. It is part of the research project "VirtuGrade" funded by the "Ministerium für Bildung und Forschung Baden-Württemberg" within the program "Virtuelle Hochschule Baden-Württemberg". The VirtuGrade project has been funded from 04/1998 to 03/2003. The course "ACL" is offered since summer 2000. The last course was taught in summer 2003. It fell victim to the transformation of our linguistics-centred M.A. course of studies to a computational linguistics-centred international B.A./M.A. program. In the ACL course, graduates from different German universities are coached by three tutors from Tübingen. About 10-20 students take part in the course each term. The course is an example of problem-based teaching. It is based on an application, an intelligent dictionary access tool, and students are supposed to understand and use the language technological basics of this tool. The learning environment MOST is used for asynchronous cooperation. MOST is an open source database developed by the Multi-Media-Lab of the University of Tübingen. The text-based open source chat tool with integrated whiteboard which is used for synchronous communication and collaboration is TULKA developed by the Department of Mathematics at Tübingen University. TULKA has been developed by two Czech students on behalf of the Institute of Mathematics, University of Tuebingen. It is distributed through the open source platform "freshmeat".

The second online seminar which we describe in this report is an Introduction to Computational Lexicography (CoLex) aimed at undergraduate and graduate students. It is part of the MILCA project funded by the „Bundesministerium für Bildung und Forschung“ (BMBF) within the program „Neue Medien in der Bildung“. MiLCA has been funded from 08/2001 to 06/2004. The course was conducted for the first time during the summer term 2002. As participants there were eight students from the Universities of Tübingen and Bochum. ILIAS was used as an asynchronous open source learning environment in this course. The text-based chat tool was again TULKA. In a second phase, students from several German universities as well as students from Hungary, China and Israel took part in the course. The course language was changed from German to English, a decision which was also due to the internationalisation of our program.

A third course in Information Retrieval has been taught by a colleague of us in Tübingen between 2000 and 2003, in collaboration with a teacher at Tilburg university. TULKA has also been used in this course, and some of the suggestions from this course have influenced the functional design of the tool.

3. Media-didactic framework

The online courses are centred around two scheduled chat sessions per week. These sessions correspond to traditional face-to-face seminar sessions and they last about 90 minutes each. During this time, the students work in plenary sessions as well as in small working groups of about 3-4 participants. For this purpose, the chat tool provides an online classroom as well as several chat group rooms. The chat sessions are structured rigidly. At the beginning, organizational topics, questions about last week's session and announcements are of major interest. After that, the teacher of the course gives an "advanced organizer" of how today's session will be structured. This part is followed by discussing and illustrating the topic of the week. During this phase, the teacher triggers discussions and questions by confronting the students with controversial arguments, case studies or central questions. The emphasis is clearly on interactive discussion and not on pure content presentation. After that, the students are sent to their respective group rooms, where they start to discuss, subdivide and assign the tasks of a group exercise which they have to solve during the rest of the week. The teachers are available for questions during the group phase, but they do not intervene actively. Finally, all of the students meet again in the online classroom, where they can ask questions, which are of general interest. Eventually, the teacher gives a short prospect of the next online session.

The asynchronous work, i.e. the autonomous preparation and repetition of the course topics, is done individually on the basis of an online course book, the so-called "hyperbook", including demo programs and additional material. Also the protocols of the chat sessions are a

valuable source of reference for the students and sometimes a starting point for a more in-depth discussion of some topics. The learning material is available via the asynchronous learning environment. On the other hand, the students solve the group exercises either asynchronously by using ILIAS resp. MOST as a facility for exchanging solution sheets and sending messages to the newsgroups. They may also decide to work synchronously by using the chat tool for discussing the tasks. In both cases, the sample solutions to the exercises eventually are made available to the students through the asynchronous learning environment.

4. E-Learning software

The following section describes the tools and functions of the multi-media learning environments of the two virtual courses described in this paper.

Figure 1 shows the area in MOST for asynchronous communication and cooperation in the ACL course. Students can upload and download files and edit messages within the newsgroups. From the content menu, they can choose between a personal space, a group area and a public area called "community".

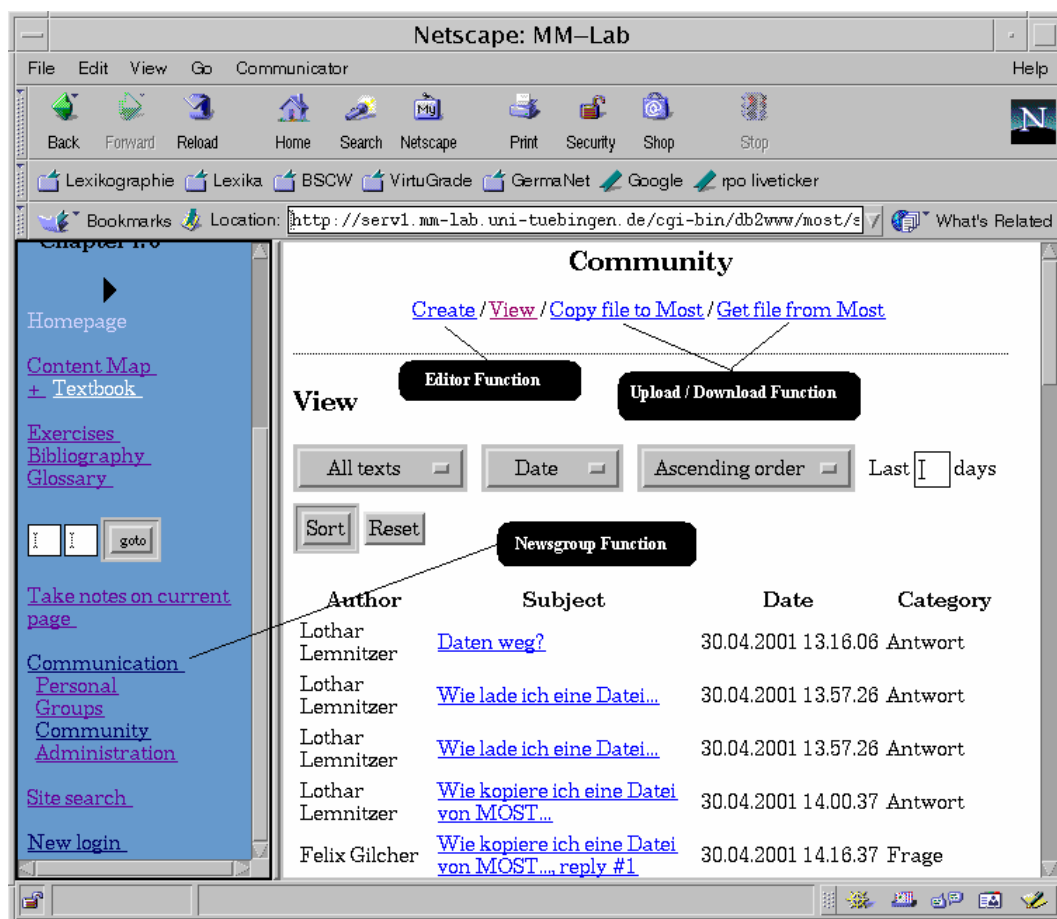


Figure 1. Asynchronous communication in MOST

The group space in ILIAS for asynchronous communication and cooperation in the CoLex course including a folder for the chat protocols can be seen in figure 2.

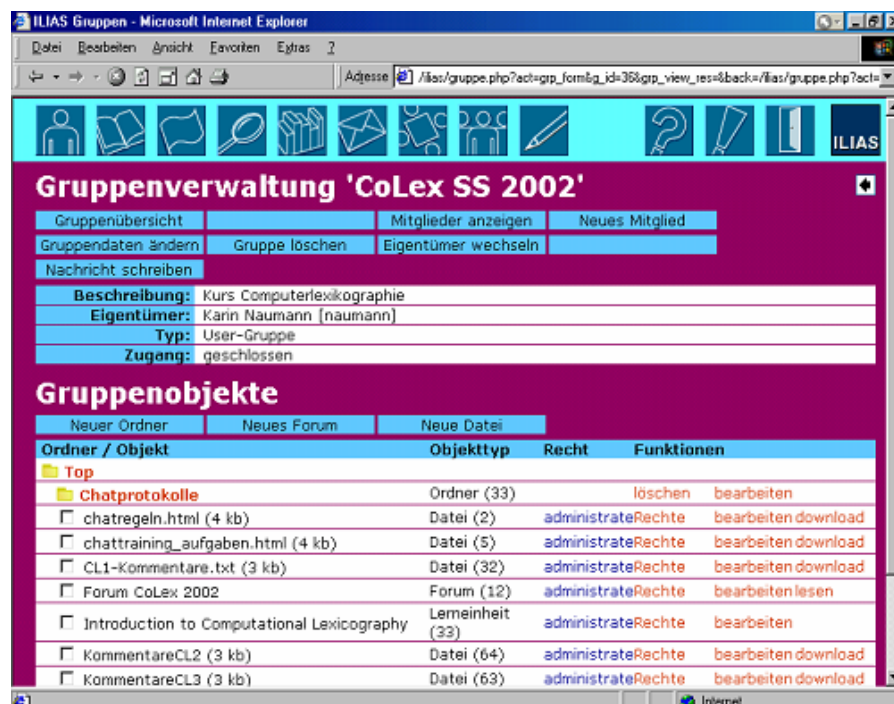


Figure 2. Communication in ILIAS

Figure 3 shows our text-based TULKA chat tool which includes an integrated whiteboard application. On the left there are several functions for drawing and writing on the whiteboard. It is also possible to upload files into the whiteboard. On the right side of the window there are functions for choosing between the uploaded files, for clearing the whiteboard, for changing from chat classroom to chat group rooms and vice versa. On this side of the browser window, there is also a list available displaying the login names of the participants present in the chat room. It is possible to open up a separate chat window.

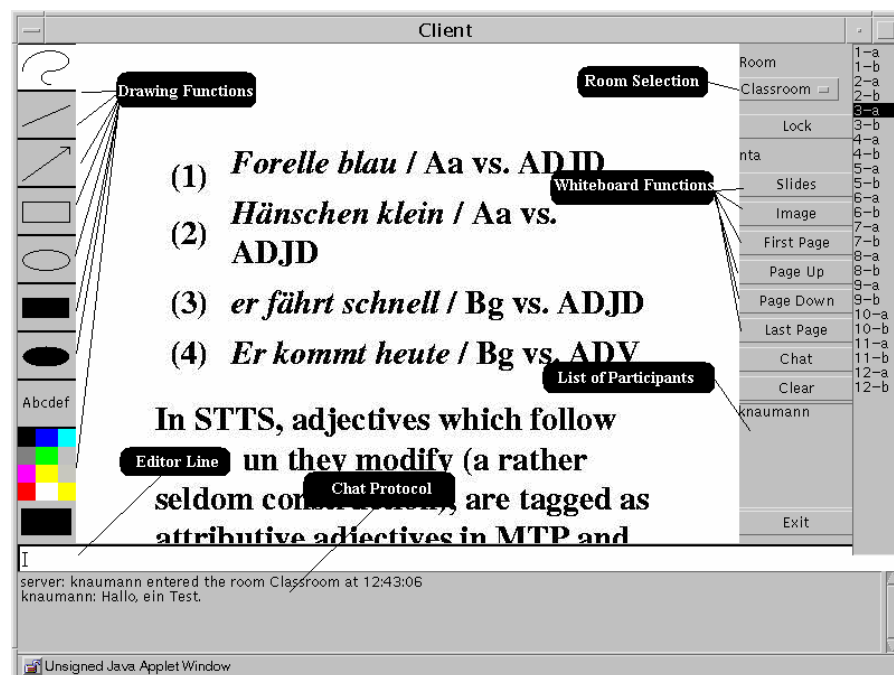


Figure 3. TULKA chat tool with integrated shared whiteboard

Figure 4 shows the additional chat window which displays different colours for different types of messages (note: server messages appear in blue font colour, messages of teachers are red, students' contributions are displayed in black colour). From this window it is also possible to choose between a public communication channel and a private "whisper" channel which is only available for a specified group of participants, in our case the teachers of the course.

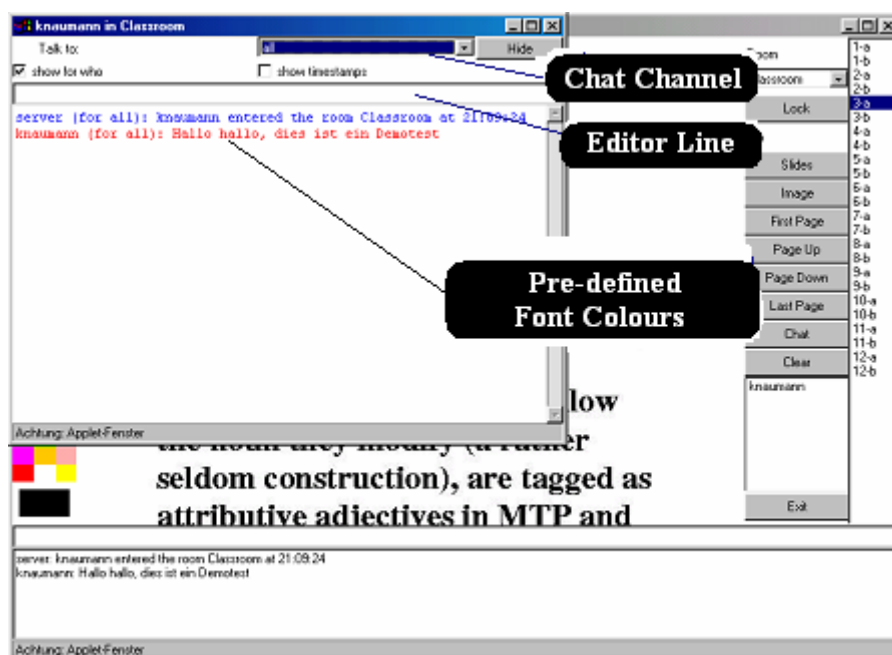


Figure 4. TULKA with additional chat window

5. Adjusting the learning environment

One of the central tasks in both projects has been to select an adequate eLearning environment, adequate for our target groups and for the subject matter which should be taught. Tool development was out of the scope of both projects, as was the purchase of commercial software. However, with the institute of mathematics developing and teaching an international course for postgraduate students, we found a partner who was willing to invest into the development of an adequate tool for synchronous communication which meets our didactic requirements.

Our initial learning environment, partially determined by project-wide decisions in the VirtuGrade project, consisted of:

- MOST - a homegrown learning management system used as a repository for our hyperbook (developed by the MultiMedia-Lab of the University of Tübingen, but not longer available)
- BSCW as a platform for document exchange and asynchronous communication
- EveryChat, a freely available, web-based chat tool.

The use of three different tools led to a cognitive overload and confusion with our students, which is clearly reflected in the evaluation of the course. (Note that both the VirtuGrade and the MiLCA project included an independent evaluation task which was performed by the Knowledge Media Research Centre (KMRC) Tübingen.) Furthermore, the chat tool proved to be inadequate for instructional purposes, as we will show later on. Student feedback as well as our own experience as teachers led us to the following decisions:

- We stopped using BSCW and integrated the communication facility into the MOST platform. This was done by the MOST developers team on our request and with our advice.
- We looked for a chat tool which was better suited for didactic purposes and found the

TULKA chat tool with integrated whiteboard, which was subsequently adapted to our needs.

- As it was foreseeable that the MOST platform would no longer be supported after the end of the VirtuGrade project, we changed to ILIAS, an open source learning management system. The system had gained, after more than four years of development, some maturity and is still under development.

In the following we want to focus on design decisions in the context of the chat tool and ILIAS. Both design decisions are a reaction on the feedback of the students and authors who participated in the projects.

5.1 The TULKA chat tool

Our teaching experiments with the EveryChat chattool and an early version of TULKA led us to specify minimum requirements for a chat tool in a didactic context:

- it should be easy to use from every workstation, i.e. web-based and platform-independent
- several chat rooms should be available for plenary meetings and group meetings
- every participant should see the protocol of the whole session, no matter when they enter the chat room. This is of particular importance when students are asked to work in group rooms and come back to the main room later
- a whiteboard should be available for everyone to present something on the spot as well as for displaying slides
- students should be required to log in with a user ID and a password, so that their chosen names are stable over the period of the course
- there should be only one public channel. The whisper channel which is well-known from many chat tools and commonly used for private conversations should be reserved for the teachers
- some groups of users and their respective rights should be definable, at least for teachers, students and guests. Different colours should be assignable to the texts of the different groups

A more detailed list of requirements is presented in Lemnitzer/Naumann 2003.

These technical functions of the chat tool, together with a chat training and a chat etiquette, turned out to be the key to successful synchronous communication in the seminar. Some features that we integrated additionally, however, turned out to be contra-productive:

In one version of the chat tool we disallowed the sending of messages without having the explicit permission to “speak”. Technically, the channel was locked exclusively for one sender. Students could apply for the right to talk, but they were not able to interrupt a speaker and could only wait to be called. This technically enforced policy turned out to be too restrictive and was turned down by the students. We therefore removed this feature. A didactic solution to prevent too much parallel threads of conversation turned out to be more successful.

In early versions of the tool everyone was allowed to manipulate the content of the whiteboard. This led to situations where students accidentally or deliberately removed contributions of other students by “clearing” the board. In later versions, the facility to lock the whiteboard has been added. The same policy has been followed with the permission to change the text colour. In the classroom, no one is allowed to change the colour. In the group rooms, however, everyone can choose their preferred colours for their contributions.

5.2 The ILIAS platform

In 2002, we decided to change from MOST to ILIAS. The ILIAS Learning Management System allowed us to compile repositories of learning objects and enabled the students to communicate and to collaborate asynchronously. The authors of the MiLCA project, however, did not appreciate the built-in editor. They also did not want to produce their materials in an environment without the possibility to export these data. Some of the partners had some legacy course material which they wanted their students to access through ILIAS. We therefore asked the ILIAS development team to allow for the import of learning object in a platform independent format and enriched with metadata. Our co-operation on this issue has been successful. The XML-based import and handling of learning objects is part of the ILIAS functions (ILIAS version 3).

6. Conclusion

It is helpful, in some cases even necessary to participate in the planning and realisation of eLearning tools, if you want to implement an innovative didactic design with your courses. Open source offers ample opportunities to do so. However, you always have to face the situation that the tool you use and adapt disappears from the market. On the other hand, this may also happen with commercial products. With a widespread open source solution, it is guaranteed that the community will profit from your contribution. Your students will be happy to enjoy a non-standard teaching and learning style, and in some cases they can even be motivated to participate in the development of the learning environment. Eventually, do not underestimate the satisfaction you gain from the feeling that you contribute to provide something which helps other people in their teaching and learning.

7. References

Lemnitzer, Lothar & Karin Naumann (2001). "Auf Wiederlesen !" - das schriftlich verfaßte Unterrichtsgespräch in der computervermittelten Kommunikation. Bericht von einem virtuellen Seminar." In: Michael Beißwenger (eds.): "Chat-Kommunikation - Sprache, Interaktion, Sozialität & Identität in synchroner computervermittelter Kommunikation. Perspektiven auf ein interdisziplinäres Forschungsfeld" (pp. 469-491). Stuttgart: ibidem.

Lemnitzer, Lothar, Naumann, Karin & Peter Zentel (2001). "Im Wortlaut - Didaktisches Design eines virtuellen Seminars mittels synchroner cvK". In: Wagner, Erwin & Michael Kindt (eds.): "Virtueller Campus: Szenarien, Strategien, Studium". Medien in der Wissenschaft, Band 14 (pp. 128-136). Münster: Waxmann.

Naumann, Karin & Lothar Lemnitzer (2002). "Lernen im Chat: ergebnisorientierte Kommunikation in einem synchronen Virtuellen Seminar". In: Rinn, Ulrike & Joachim Wedekind (eds.): "Referenzmodelle netzbasierter Lehrens und Lernens. Virtuelle Komponenten der Präsenzlehre" (pp. 201-219). Münster: Waxmann.

Lemnitzer, Lothar & Karin Naumann (2003). "Lernen von der Bettkante aus" - Der virtuelle Kurs Applied Computational Linguistics. In: "Sprache und Datenverarbeitung", Jahrgang 27.

Lemnitzer, Lothar & Karin Naumann (2004). "Ein überregionales rein virtuelles Lehrangebot zur angewandten Computerlinguistik." In: Zentel, Peter & F.W. Hesse (eds.): "Netzbasierte Wissenskommunikation in Hochschule und Weiterbildung. Die Globalisierung des Lernens" (pp. 101-121). Bern: Huber.

Naumann, Karin (2005): "Kann man Chatten lernen? - Regeln und Trainingsmaßnahmen zur erfolgreichen Chat-Kommunikation in Unterrichtsgesprächen". In: Beißwenger, Michael & Angelika Storrer (eds.): "Chat-Kommunikation in Beruf, Bildung und Medien: Konzepte - Werkzeuge - Anwendungsfelder" (pp. 257-272). Stuttgart: ibidem.

8. Internet Links

Internet Seminar „Applied Computational Linguistics“ (ACL): <http://milca.sfs.uni-tuebinge.de/ACL/release> . Last visited: 13.10.2005.

Internet Seminar "Computational Lexicography" (CoLex): http://milca.sfs.uni-tuebingen.de/B2/SS_2004 . Last visited: 13.10.2005.

VirtuGrade Project: <http://www.virtugrade.uni-tuebingen.de> . Last visited: 13.10.2005.

MILCA Project: <http://milca.sfs.uni-tuebingen.de> . Last visited: 13.10. 2005.

Research Program "Virtuelle Hochschule Baden-Württemberg": <http://www.virtuelle-hochschule.de> . Last visited: 13.10.2005.

Research Program "Neue Medien in der Bildung": <http://www.medien-bildung.net> . Last visited: 13.10.2005.

Open Source chat tool (plus whiteboard) TULKA: <http://themes.freshmeat.net/projects/tulka> . Last visited: 13.10.2005.

Open Source chat tool EveryChat: <http://www.everysoft.com/frames.html> . Last visited: 13.10.2005.

Open Source Environment ILIAS: <http://www.ilias.de> . Last visited: 13.10.2005.

Learning Environment BSCW: <http://bscw.fit.fraunhofer.de/> . Last visited: 13.10.2005.

Multi-Media-Lab, University of Tübingen: <http://www.mm-lab.uni-tuebingen.de/> . Last visited: 13.10.2005.

International B.A / M.A. program in Computational Linguistics: <http://www.sfs.uni-tuebingen.de/iscl> . Last visited: 13.10.2005.

Knowledge Media Research Center (KMRC) Tübingen: <http://www.iwm-kmrc.de> . Last visited: 13.10.2005.

Towards the open cross-disciplinary research environment

Ernesto Damiani¹, Paul G. Mezey², Paolo M. Pumilia³, Anna Maria Tammaro⁴

¹. Dip di Tecnologie dell'Informazione, Milan Univ; ². Canada Research Chair in Scientific Modelling and Simulation, Memorial University of Newfoundland, St. John's, NL; ³. Chair of the 'Open Culture' promoters group; ⁴. Dip dei Beni Culturali, Parma Univ.

Abstract

This paper outlines some contemporary theoretical and technological issues that are becoming of paramount importance for building a cross-disciplinary research and knowledge sharing environment. Emphasis is placed on three key elements of Open Culture: interoperability and open assets, re-use and intellectual property rights management and quality assurance.

1 Introduction

The spreading of ICT in a variety of contexts is raising the urgent need for a transformation in how we experience and exchange knowledge and will have substantial impact in scientific and commercial production as well as in the learning process. Lifelong learning and cross-disciplinary collaboration competencies will be more and more required by scientific institutions and enterprises, while knowledge sharing will become the *sine qua non* of collaborative research.

Boundaries between research areas are increasingly blurred and growing numbers of investigations are being carried out by larger and geographically distributed teams, composed of different kinds of professionals working in close contact.

The ability of sharing knowledge is getting more and more important in universities. The patterns and cadences of interaction among faculty members, learners, instructional development staff, knowledge management staff, and expert practitioners will assume new forms. The ability of generating *just-in-time knowledge* will spread in parallel with a decline of the relative importance of static knowledge. Pervasive, perpetual learning, richly supported by knowledge management, will become the new "gold standard" for many learners' experience [Collier2003].

It is clear that we cannot naively rest and merely rely on the power of the ever more advancing new technologies to successfully face such challenges.

More efficient organizational frameworks for the production of knowledge components and for their sharing and reuse are to be devised and fine-tuned, together with aptly suited international laws to regulate authorship management and a stronger commitment to manage and assess the quality of information. But the establishment of such infrastructure will not be enough. Ability to create and maintain new relationships with co-workers, often outside the institutional context, and the attitude to emphasize truthfulness of such relationships should be tied with the above changes, thus positively affecting the whole spectrum of interpersonal relations and driving the same kind of thorough cultural change that can be already witnessed by some successful open source software projects.

We all have to re-educate ourselves: the culture of open source places very high new demands on data quality, information quality, and source code quality, as well as on accountability of all those contributing code and data, as well as of those who merely use the benefits of open source. These traits countersign the new phenomena of open access repositories and open content for e-learning as well as traditional open source software development, although rules governing the conduct of participants are still evolving and somewhat unclear.

In our opinion, this process will trigger cultural changes leading to new demands for higher degrees of responsibility to everyone, and, in the fullness of time, to a new, cross-cultural code of *Internet Integrity*. In other words, the most profound effect of *open culture* is the actual development of a new culture, incorporating the needs for openness, as well as the requirements of a new level of 'integrity'. Openness provides the benefits, whereas the new levels of integrity provide some of the safeguards against misuse of the powerful tools involved.

We argue that such open culture will have greater impact in the interdisciplinary fields, where a certain lack of familiarity of experts in one field with the subject matters of the other fields is

the ground for a natural caution, slowing down both communication and progress. Thanks to the increased levels of trust generated by the open culture approach, these difficulties will be reduced.

This demands a proper attitude by researchers, as well as proper tools to help researchers to grasp concepts when reading papers or following seminars in different fields. Much of the actual developments are still ahead of us, and by extrapolation from the developments of the recent past, the effects of attitude changes originating from open cross-disciplinary interactions may provide for many surprises.

Although such exciting turmoil, promising great opportunities and bringing out new business models, scholars are often unaware of such possibilities or unable to exploit them properly and the standing commercial interests are often opposing the trends, denying, restricting or delaying access to publications, especially to students and to researchers from the less rich countries.

This paper outlines some contemporary theoretical models and discusses emerging technological platforms in which the issues of re-use and integrity are becoming of paramount importance. Emphasis is placed on the three key elements of the Open Culture: - interoperability and open assets; - re-use and intellectual property right management; - quality assurance and integrity.

2 Open Culture

Different communities are growing, belonging to open source (software with available source), open access (based on interoperability of digital resources), open content (easy-to-deploy and re-use of learning material), sharing the belief that knowledge cannot be considered as exclusive property of those who made the discovery and that inventors will benefit, in terms of feedback from peers and also financially, by spreading out the obtained results, under the least restrictions as possible, while benefiting from the international laws to preserve authorship [Boyle2003].

There is room for plenty of synergies among those communities, once the common ground for acting *openly* is firmly established. What does it mean to be open? Originally, open has meant free sources available for the open source community, the oldest and most organized of the participating communities. Open means interoperable for the open access community, started with open archives experiences and representing an alternative or subversive proposal to the current scientific publication structure. Finally open means encouraging reusability and free spreading of digital resources for improving learning and teaching for the e-learning community.

'Open Culture' appears as the unifying perspective around which the different communities can meet, as Lawrence Lessig claimed before [Lessig2004]. Echoing the "global brainstorming" and cultural change ideas put forward by Paul Mezey (Open Source Contents Workshop, Didamatica 2004 Conference, Ferrara University, Ferrara, Italy, 2004 May 10-12, invited lecture, Paul G. Mezey: Open Source: The Goals, the Tools, and the Culture), an intriguing description of that concept has been recently given by Mark Hemphill[oc2005]:

Open Culture is like a cross disciplinary brainstorming. Sharing information and knowledge stimulates cross disciplinary interactions and gives additional significance to individual knowledge. Before Internet, there was the practice of brainstorming, now Internet and communication methodologies offer a new level of integration. Sharing and exchanging knowledge is good not only for me, as an individual but for the entire society, however there is a cultural resistance. Sharing knowledge throughout Internet should have the same integrity as in interpersonal communication.

Is also worth noting that there are differences between science and humanities communities, as the latter is more individualistic than the former. In any case, change agents are needed to move people to the culture of knowledge sharing. As Fred Friend puts it [oc2005], referring to the 'open archives':

If they do not know how, we can address this problem? Key questions are: who owns the fruit of academic research? And what is the value of dissemination of research results (often publications)? What libraries pay?

The required cultural changes will lead to new demands for higher responsibility by everyone, and to a new, cross-cultural code of Internet Integrity. The more profound effect of open culture is the actual development of a new culture, incorporating the needs for openness, as well as the requirements of a new level of 'integrity'. Openness provides the benefits, whereas the new levels of integrity provide some of the safeguards against misuse of the powerful tools involved.

3 Interoperability, integration and decentralization of repositories

In a technical sense, *open* means first of all *interoperable*. In the future this will include the whole complexity of integration not limited to protocol and standard. There will be considerations of the possibility of multiple choices and options, from simple adoption of OAI-PMH protocol and metadata indexing to ontology driven integration and to content publishers (Digital Asset Management Systems, portals, content management systems and unstructured data-handling solutions).

The convenience of small scale open access archives has been explored and the evolving structure of distributed repositories and independent services, as sources for automated data search and aggregation has been vastly considered by the Open Access Initiative (OAI) since 1999.

The OAI has been a first step for developing the architecture of information and knowledge sharing, distinguishing data providers and service providers on the net. The OAI-PMH (Protocol Metadata Harvesting) represents the core requirement for interoperability and the unifying link between the two communities.

No simple solutions are available and there are many factors to be taken into account for new scholarly communication using OAI-PMH. Ultimately, the opportunities and risks of the OAI Initiative, the good balance between commercial service providers and cultural institutions, content providers and their different points of view, should be better analyzed and determined to build an efficient way of academic publishing.

Currently, the OAI is representing a revolutionary threat for the conventional academic world. In this regard, an interesting example is evidenced by an English project [Jeffery2005]. Linking CRISs (Current Research Information Systems) database and OA (Open Access) Systems brings together systems for managing research and development (R&D) in universities with systems for providing open access to scholarly publishing. The major visible outputs of R&D and scientific publications are using the emerging European GRIDs infrastructure.

The debate over OAI is very active, with the components 'green' (institutional and thematic repository for self-archiving) and 'gold' (author / institution pays publishing at publishers' server) as competing but also complementary processes. The project gives evidence that the scientific process can be treated as a workflow, with recording of outputs at various stages, from initial research ideas to project proposals to interim reports and final publications - along with the produced data, software and cross-references to other works. The knowledge base considered by the project consists not only of the white literature (publications which have had a formal publication process) but also the 'iceberg' of grey literature, encapsulating the know-how of the organization in technical reports, instruction manuals, training materials etc. Furthermore, increasingly, the information process and the knowledge base rests in datasets (e.g. results of drug clinical tests), in databases (e.g. customer relationship information) and in software (which encapsulates much of the business processes of the organization).

The consequences and the impact of interoperable repositories and common access services could be very important for a change in scholarly communication. For example, research quality could be measured not only as publications produced and other biblio-metric indicators. The project CRISs provides both a context for evaluation of, and understanding the background to scholarly publication. CRISs also provide a management framework for R&D in academic institutions from funding agencies through national laboratories to universities, as well as a mechanism for interoperating research and development information.

4 Re-using and Creating New Knowledge for a multidisciplinary community

While the crisis of access to digital content and the difficulties associated with rights management are some of the most pressing issues in cyberspace today, the open content appears as a new, although still somewhat fuzzy, concept coming to the rescue by prompting the search for peculiar balances between the need for access versus control. Generally speaking, the open content would allow copying of content without restrictions; in some way authors might provide for content modification, as well. The public domain is to be included as a special open content case.

For this vision of world-wide access to information to become a reality, knowledge must be created, organised and stored in formats and architectures accessible to everyone [Mason2005]. The author proposes a conceptual framework for thinking about knowledge management in the context of digital libraries that may serve multiple cultures. The framework is grounded in the context of boundary spanning, a concept that acknowledges the need for mechanisms for communication across the boundaries between domains of knowledge and experience.

Digital libraries are trying to identify barriers and obstacles to knowledge sharing, while paying attention to preservation issues, highlighting possible mixed business solutions involving open content creation, also recognizing success factors for using and re-using open contents besides the target community and looking for agreements with common quality criteria and definitions of rules involved for creating collaboration frameworks.

In the framework of digital libraries, knowledge creation and re-using of content have three main types: community-centred, product-centred, and systems-centred.

- **Community centred:** Rather than focusing solely on the individual user who interacts with open content, digital libraries consider the group, organization, and community activities and concerns which give rise to information-related behaviour. There is evidence of more convergence of information and communication technologies, blurring the lines between tasks and activities and between knowledge creation, e-learning and digital content. This trend extends not only to commercial content but also into the open content of cultural heritage institutions wishing to benefit from digital content, in such scenarios as e-learning, e-government, cultural tourism.

For Nancy van House digital libraries should support the cognitive work, bringing to creation of new knowledge and facilitating learning. Cognitive work is characterized by three elements:

1. It is situated in a cultural context;
2. It is distributed;
3. It is a social work.

Among others, Wenger [Wenger1998] spoke about the theory of community of practice and community of interest aimed at knowledge work in a particular context.

- **Product-centred:** digital collections, open digital content, the Web, e-commerce developments together with DRM systems or other suitable legal framework, combine to create myriad opportunities for repurposing content into diverse distribution channels, to maximize revenue for service providers but also for re-using content for learning and teaching purposes. This theme focuses on digital document granularity and format, including metadata, bibliographic control and semantic indexing.

According to Lynch [Lynch2000], enabling the identification of digital works is not the only purpose of bibliographic control, but it is certainly one of the most important and most widely relied-upon aspect. But the practices of information finding are changing in a world of digital information and computer-based search systems. The real revolution in access is going to be driven by the availability of massive amounts of content directly in digital form rather than print, and by the emergence of network-based computer systems that provide an environment not just for identifying content (which historically existed in print form and was used offline, independent of systems like online catalogues) but for its subsequent actual use and analysis within the access system.

- *Systems-centred*: open culture expand the interest of digital repositories in information storage and retrieval to include preceding and succeeding phases, incorporating the processes of creating, using, and disposing of information. The trend toward knowledge management, as an overarching learning architecture philosophy and methodology, is evidenced in the myriads of technological artifacts, such as digital repositories and Learning Content Management Systems (LCMSs), which have emerged to capture, categorize, and manage digital instructional content or learning objects.

Diaz V. and McGee P [Diaz2005] identify the need to examine existing knowledge management models from a planning and decision-making perspective. They discuss four current models of knowledge management found in higher education:

1. the *traditional model*, they call 'pre-digital' which is teacher centered and based on textbooks;
2. the *intellectual capital/appropriative model*, called 'Intellectual Capital', characterized by the education as a market product or process and by the diversity of students and where the Intellectual Property Rights govern the systems (using Learning Management Systems as WebCT, etc.),
3. the *sharing/reciprocal model*, where the prevailing organizational form is that of a Consortia of educational institutions, controlling Intellectual Property Rights with licenses like Creative Commons and based on networked directories of Learning Objects and institutional repositories distributed worldwide like DSpace (see Section 5)
4. the *contribution/pedagogy model*, based on the requirement of a learning community sharing its knowledge, i.e. moving on from the transmission to the cooperative creation of knowledge, for example with tools as Wiki, OSCAR, etc.,

They propose a new, relativist model of knowledge management that accommodates cross-institutional cultures and beliefs about learning technologies, construction of knowledge across systems and institutions, and the trend toward learner-centred environments, disaggregated and re-aggregated learning objects, and negotiated intellectual property rights. Further, they examine and showcase institutional instances of various knowledge management models and propose the Open Knowledge Model, developed to address learner-centered environments.

5 Community-based Knowledge Sharing Platforms

In the scenario of the knowledge-based society, communities are emerging as a new organizational form supporting knowledge sharing, spreading and application processes. Communities do not operate in a vacuum; rather they have to confront with a huge amount of digital information, such as text or semi-structured documents in the form of web pages, reports, papers and e-mails. Experience has shown that basic communication and data processing technology is not enough to support community-based knowledge sharing. The capability of extracting and handling classifications of heterogeneous documents produced by multiple sources is an essential requirement for information sharing; it can dramatically improve the effectiveness of community-wide cooperation. From an architectural point of view, a community-based knowledge-sharing platform is composed of two main parts: a *knowledge interchange infrastructure* and a *meta database*. Recent research and development has proposed a number of architecturally diverse platforms, going from service-oriented distributed architectures like DSpace, CDSWare and Ariadne to "pure" peer-to-peer ones like Edutella [Nejdl2003].

Regardless of the architecture, metadata plays a crucial role when it comes to the integration of existing infrastructures. For our purposes, we distinguish three categories of metadata:

- 1) *Descriptive metadata*: the metadata elements used for describing resources, i.e. knowing what they are and what they are about. These include the domain model (e.g., a shared ontology) used as a vocabulary for descriptions.
- 2) *Technical metadata*, describing the internal operation and behavior of the knowledge sharing platform
- 3) *Administrative metadata*, used for managing the information exchange, e.g. controlling access to resources and their level of use. Administrative metadata also include licenses and other intellectual property-related information.

Multiple attempts at standardizing general metadata have been independently made both by the e-learning and the digital libraries communities. The latter have a much longer history and resulted in the development a number of heavyweight standards like Z39.50, aimed at large scale cataloging of non-digital resources. More recently, attention focused on lightweight metadata standards aimed at the description of digital objects, with the aim of supporting, besides learning, creative innovation and research. For the purposes of this Section, after a brief review of available standards for knowledge-description metadata, we shall take as a reference the metadata and the technical infrastructure chosen for the DSpace repository to be used as a pilot implementation within the project *European meta database of E-Academic resources* (EUREA).

Dublin Core is a library-oriented standard aimed at defining a bare minimum (*core*) set of metadata elements (such as author, title, etc.), for digital libraries' cataloging and interchanging purposes. Special profiles (e.g., the Education Profile) have been defined for adding domain-specific elements to the DC standard.

MARC (Machine Readable Cataloging) is the metadata format originally defined for the U.S. Library of Congress. It provides a mechanism for computers to exchange and process bibliographic data. There are several variations of MARC, the latest being MARC 21. This standard divides each bibliographic record into logical fields (like author, title, etc.), in turn subdivided into subfields. The repertoire of MARC fields and subfields is encoded as a set of numerical tags whose meaning are spelled out in MARC documentation. The MARC format is semantically rich and widely used by the digital libraries community. However, it is aimed at the specific institutional purposes of a librarian (classification and conservation) rather than at the free interaction style of research communities.

IEEE-LOM (Learning Object Metadata) is a standard published by the Institute of Electrical and Electronic Engineers (IEEE). It is composed of multiple parts, including a data model describing a set of elements with a defined semantics. These elements are grouped into nine categories: General, Life-Cycle, Meta-Metadata, Technical, Educational, Rights, Relation, Annotation and Classification. These nine categories, forming LOM *base schema*, were designed as a refinement of the three main ones of descriptive, technical and administrative metadata introduced above. The LOM model is a hierarchy of elements, whose leaves are either *simple* (i.e. containing a single value) or *aggregated* (containing multiple values). Each leaf element has a name, an explanation, a size, an order and a usage example. This complex structure was designed to be machine- rather than human-readable and is aimed at fostering interoperability and free interchange of heterogeneous objects between platforms. LOM intricacies discouraged developers from attempting full-fledged implementations of it, although many partial ones exist

SCORM (Sharable Content Reference Model) is a standard developed by the U.S. Department of Defense for technology-based learning across the federal and private sectors. SCORM deals with Sharable Content Objects (SCOs), i.e. a collection of multimedia assets (image, text, sound) that becomes an instructional unit. SCORM metadata schema is based on LOM's nine categories.

Of course current technological platforms for knowledge exchange do not support all these standards; for instance, at the moment DSpace only supports the Dublin Core Metadata Element Set, plus some extensions conforming to the DC Library profile. In other words, items available for exchange in a DSpace all have a single Dublin Core Metadata record. The development team behind DSpace at MIT has announced its intention to support a subset of the SCORM element set within DSpace in the year 2006.

An important and often neglected step in metadata lifecycle is *harvesting*, i.e. metadata collection and processing by software agents (*harvesters*), e.g. to create custom views and, more importantly, registries and cross-references. As mentioned in Section 3, *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH) provides an application independent interoperability framework for metadata harvesting which collects metadata from different repositories and creates an *open registry* where each metadata entry links back to the original metadata item. From an architectural point of view, main assumption behind OAI-PMH is a clean separation between data and service providers. Data providers establish an OAI-PMH-based interface to their own local digital resources, while service providers collecting and integrating metadata from multiple repositories. Harvesters like ARC [Liu2001] provide

facilities for searching across multiple archives plus other value-added features like resource ranking. OAI-PMH supports dissemination of multiple metadata formats, but for interoperability purposes it mandates dissemination of Dublin Core without additional profiles. Mapping complex metadata formats to unqualified Dublin Core may be difficult or even unfeasible without severe loss of semantics.

In service-oriented architectures like DSpace, metadata are treated differently from ordinary data, especially as far as modification permissions are concerned. In DSpace's metadata registry, entries can be edited only via a custom administrative interface to be used at certain pre-set steps of the Dspace population workflow. The workflow process in DSpace is supporting self-submission of digital objects (e.g., learning objects or scientific "grey literature" items) on the part of their authors, but monitor the submission process in order to guarantee that the objects' self-descriptions will be standard enough to support effective query and retrieval.

The rationale behind this choice is that metadata cannot be treated lightly: they map data items to the shared conceptual space, and changing them freely would inevitably bring in errors, potentially even loss of resources ending up associated with the "wrong" metadata elements. Also, even on occasions when Dspace users are allowed to modify the values of the Dublin Core description fields, in principle they should use only terms from a shared, controlled resource classification.

Technological platforms and their architecture are not neutral with respect to the modalities of knowledge sharing. In a recent paper [Stuckenschmidt2005], W. Nejdil argues that the combination of advanced metadata and P2P technologies is suited to deal with most problems of inter-organizational knowledge management; his research group has also presented search techniques specifically aimed at distributed research communities [Chirita2005]. Concrete applications and scenarios, including cooperative research, have specific requirements and constraints that may require ad hoc design decisions. A crucial point is the role shared vocabularies and ontology play in these different areas. Service-oriented architectures support a shared metadata structure aimed at integrating different point of views in a common domain model, e.g. providing the community with common description vocabulary or domain ontology. The peer-to-peer approach tolerates (and indeed encourages) a certain degree of inconsistency among individual domain models, in order to effectively support a high rate of change. Inconsistency between local models and vocabularies will eventually be resolved via continuous interaction among community members.

A second major aspect where the choice of a service-oriented vs. a "pure" P2P platform architecture is not neutral is metadata harvesting. As we have seen, OAI-PMH defines service relationships between the data provider, the service provider and the final clients. However, digital libraries in most cases act both as a client and as a server, trying to obtain outside material for their internal users and offering resources to other libraries. A standard OAI service-oriented architecture has a many-to-many structure: multiple service providers harvest multiple data providers. This is a potential cause of inefficiency: when a user queries multiple service providers the results might overlap, and duplicates handling will become necessary. Also, conventional the service-oriented architecture is prone to the well-known "cold-start problem": new sources of knowledge may find it difficult to attract the attention of a service provider, be harvested and have their voice heard.

In a "pure" P2P harvesting system [Ahlborn2002] there is no separation at all between service and data providers: each peer maintains two separate sub-systems for metadata storage and query handling. Of course hybrid approaches are also possible: even P2P networks may require additional service providers, which replicate metadata, greatly enhancing the system's overall performance and reliability.

Finally, the underlying technical platform is not neutral with respect to *Intellectual Property Rights* (IPR). The protection of intellectual property was introduced a long time ago before any computer or even electrical device was invented: IPR are rights granted to persons and aimed at protecting their intellectual creations, whether technical, scientific or artistic. Copyright grants to the author of such material a right to control the exploitation of her work, including the right to allow/disallow reproduction and any other form of public communication; therefore is a central entity in any knowledge exchange infrastructure. A key requirement for any platform is supporting some form of Digital Right Management (DRM) avoiding unauthorized use of protected multimedia contents, while supporting integration and use of open objects.

While some "open" objects may well (and usually do) adopt their own access policy, e.g. one imposing some limits to commercial secondary uses of their content, the DRM approach appears to be in conflict with the 'open content' features, as summarized in the previous section. However, creative exploitation of such technologies cannot be completely ruled out, in principle. As Weber noticed [Weber2005], the open source movement's distinguishing characteristic is its concept of intellectual property, which is centered on "the right to distribute, not the right to exclude."

The European Union's 2001 Directive on Copyright in the Information Society has tried to provide a list of exceptions to IPR protection that member states are expected to uphold. Among the list of exceptions, some look relevant to creative cooperation and learning. Two major exceptions are *illustration for teaching* and *library privileges*. The former exception wording allows free use of copyrighted material for (non-commercial) teaching purposes: however, in some countries it applies to the material's reproduction in the classroom and not to its communication to the public, which means that free teaching use of IPR-protected material on open e-learning platforms might be challenged by owners.

6 Quality issues

In a knowledge-based society the organisation of research and education systems must have the ability to contribute with quality based contents and services in accordance with changes in the world of work and society.

Specific emphasis is:

- on the development of an internal quality culture, strengthening institutional and community (as peer review) quality evaluation together with external evaluation processes;
- how to agree with common quality criteria and definition of roles involved for creating open collaboration framework (for example for e-learning, scholarly communication, etc.).

This work is based on the evidence that knowledge is culturally derived, acquired, and applied, and that learning - the acquisition of new knowledge - is enabled by skills that are culturally dependent. These cultural bases [Mason2005] for knowledge creation and absorption mean that knowledge management systems, especially those supporting digital libraries, must take culture into consideration in their design and implementation if they are to realize their potential for providing access to the widest range of knowledge.

Mason proposes a conceptual framework for thinking about knowledge management in the context of digital libraries that may serve multiple cultures. The framework is grounded in the context of boundary spanning, a concept that acknowledges the need for mechanisms for communication across the boundaries between domains of knowledge and experience. The relationship between culture and learning (the acquisition of new knowledge) suggests that knowledge management techniques that are appropriate in one culture may not be effective for digital libraries that seek to serve multiple cultures.

Boundary spanning has been recognized as a necessary component in processes that require coordination and translations among diverse groups [Star1989] and different functional groups or 'thought worlds' [Dougherty1992]. Individuals within communities of practice (CoPs) share similar experiences, a similar language, similar ways of learning, and similar values. We might extend the CoP concept to virtual communities' individuals linked through information and communications technologies? and refer to these communities as 'networks of practice' or NoPs [Brown2001].

7 A Case Study

With the help of the internet and new communication methodologies in the Information Society, we have reached a new level of integration of intellectual activities and a much broader involvement of very large number of individuals and organizations who can now participate in collective thinking, brainstorming, and the very construction as well as continuous improvement of new intellectual products, whether for science, or education, or industrial management, or entertainment.

Cross-disciplinary interactions are developing in the direction of internet-based brainstorming. While brainstorming has been a beneficial practice in much more limited settings before, today with the internet and new communication methodologies in the Information Society, we have reached a new level of integration.

Moreover, the open source and open access approaches suggests that higher and quicker results can be reached when institutional borders can be lowered a bit, thus encouraging contributions by scholars outside the acknowledged group. While this potentiality has always been realized to some limited degree, especially within the scientific communication for diffusing theoretical concepts and ideas, only with the development of the ICT has this approach a truly realistic and widespread opportunity.

As in the open source software case, exploiting such possibility is not straightforward and cannot be taken for granted. Conditions have to be met, including: i) Proposers of an exciting scientific project have to be well involved in that field; ii) they have to be able to explain their newly acquired knowledge clearly, both to colleagues and to people involved in very distant fields; iii) they have to have the ability to leverage and coordinate contributions from different scientists and scholars for the progress of the actual study.

Examples of a few meaningful cases of scientific contexts, in which the open source and open culture approach proved to be effective and where behaviour and attitude changes helping the interconnections using the open technological environment are evident, are presented in the last section of the chapter.

An example from interdisciplinary scientific modeling studies: Toxicological risk assessment of poly-aromatic hydrocarbons, using computational molecular shape analysis and plant growth monitoring:

A.Mallakin, P.G. Mezey, Z. Zimpel, K.S. Berenhaut, B.M. Greenberg, and D.G. Dixon,
Use of Quantitative Structure-activity Relationship to Model the Photoinduced Toxicity of Anthracene and Oxygenated Anthracenes, QSAR & Combinatorial Science,
in press (34 ms pages, accepted 2005 Feb.19). Published online June 1, 2005.

This study, involving plant-biologists, chemists, toxicologists, physicists, and computer scientists, is truly interdisciplinary, where the open culture elements are evident. By breaking down barriers between different fields, and bridging the differences in the focal points of result analysis and interpretation, not only the overall goals of the actual project have been met, but a mutually beneficial attitude change also has occurred, that has broadened the perspectives and also the scientific efficiency of all participants of the project. Not only the perspectives of the participants have broadened, but actual methodologies have been adopted in fields where they have not been used before, and methodologies tested and well established in one field have become accepted tools in a much broader area of science. Furthermore, this example also has indicated that broader society issues, such as pollution in the environment, can also benefit when cross-disciplinary, open culture approach is applied in science.

8 Conclusions

This paper provided a preliminary discussion of some contemporary theoretical models and emerging technological platforms, focusing on the issues of re-use, IPR management and integrity protection. Namely, we outlined how future interdisciplinary research and knowledge sharing environment will have to take into account some key elements of the Open Culture.

References

oc2005 - The Conference held from 27 to 29 June 2005 in Milan University has been the first opportunity, to the Authors' knowledge, for such diverse communities to meet. Proceedings of the Conference *Open Culture - Accessing and sharing Knowledge*, June 2005, Antonella De Robbio and Valentina Comba, Anna Maria Tamaro, Paolo Pumilia and Luigi Colazzo Editors; also available on the web as Open Archive: <http://openculture.org/milan-2005>
Boyle2003 - A thoughtful description of the impending risks over the open knowledge flowing in the digital era can be found in *The Second Enclosure movement and the Construction of the Public Domain*, by James Boyle, published on the *Law & Contemporary Problems Journal*, vol. 66, 2003, Duke Univ. The author is hoping specialized organizations will rise, aimed at investigating and divulging the emergent concept of public domain in our time. The paper can

be read at <http://www.law.duke.edu/journals/66LCPBoyle> or downloaded:

<http://www.law.duke.edu/pd/papers/boyle.pdf>

Lessig2004 - *Free culture*, by Larry Lessig, Penguin press 2004, also available on the web:

<http://www.free-culture.cc/freeculture.pdf>

Collier2003 - *Share and Share Alike: The e-Knowledge Transformation Comes to Campus*, by Geoff

Collier, Paul Lefrere, Jon Mason, Donald Norris and Robby Robson, In Proceedings of the

National Learning Infrastructure Initiative Conference, 2003; also available on the web:

http://www.scup.org/eknowledge/case_studies/ShareandShareAlike.pdf

Ahlborn2002 - Ahlborn, B., Nejd, W., Siberski, W. (2002) *OAI-P2P: A Peer-to-Peer Network for Open Archives*, Proceedings of the IEEE International Conference on Parallel Processing Workshops, 2002.

Brown2001 - Brown, J. S. (2001). Knowledge and Organization: A Social-Practice Perspective. *Organization Science* 12, 198-213.

Chirita2005 - Chirita P.A., Damian A., Nejd W. and Siberski W. *Search Strategies for Scientific Collaboration Networks* Workshop on Information Retrieval in Peer-to-Peer-Networks, ACM Fourteenth Conference on Information and Knowledge Management (CIKM 2005), November 2005.

Dougherty2002 - Dougherty, D. (1992). *Interpretive Barriers to Successful Product Innovation in Large Firms*. *Organization Science* 3, 179-202.

Jeffery2005 - Jeffery, K. G. (2005). *CRIS + Open Access = The Route to Research Knowledge on the GRID* Paper presented at the 71th IFLA General Conference and Council "Libraries - A voyage of discovery", Oslo, 14-18 August 2005, 2005.

Lynch2000 - Lynch, C. (2000). *The New Context for Bibliographic Control In the New Millennium*. Paper presented at the Bibliographic Control for the New Millennium conference, Library of Congress, 2000.

Liu, X. Maly K., Zubair M., and Nelson M. L., "ARC: an OAI service provider for cross-archive searching", ACM/IEEE Joint Conference on Digital Libraries, 2001

Mason2005 - Mason, R. M. (2005). *The Critical Role of Librarian/Information Officer as Boundary Spanner Across Cultures. Humans as Essential Components in Global Digital Libraries*. Paper presented at the 71th IFLA General Conference and Council "Libraries - A voyage of discovery", August 14th - 18th 2005, Oslo, Norway, 2005.

Nejd2003 - Nejd, W., Siberski, W. and Sintek, M. (2003), *Design issues and challenges for RDF- and schema-based peer-to-peer systems*, SIGMOD Record, September 2003.

Nejd2002 - Nejd, W., Wolf B., Staab S., and Tane J. (2002), *EDUTELLA: Searching and Annotating Resources within an RDF-based P2P Network*, International Workshop on the Semantic Web, WWW2002.

Star1989 - Star, S. L., and Greisemer, J. R. (1989). *Institutional Ecology, 'Translations' and Boundary*

Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39.

Social Studies in Science 19, 387-420.

Stuckenschmidt2005 - Stuckenschmidt H., Siberski, W. and Nejd, W. *Combining ontologies and peer-to-peer technologies for inter-organizational knowledge management*. *The Learning Organization* 12(5), 2005

Wenger1998 - Wenger, E. (1998). *of practice: learning, meaning and identity*. Cambridge University Press, Cambridge.

Diaz2005 - Diaz V. & McGee, P. *Developing Functional Policies for Learning Object Initiatives*.

Planning for Higher Education. Diaz, V. & McGee, P. *Distributed learning objects: An open knowledge management model*. In A. Metcalfe (Ed.), *Knowledge Management and Higher Education: A Critical Analysis*. Hershey, PA: Idea Group Inc.

Weber2005 - *Open Source, Open Market for Ideas*, interview to Steven Weber, professor of political science and director of the Institute of International Studies at the University of California at Berkeley:<http://www.ciainsight.com/article2/0,1540,1837245,00.asp>

Stud.IP – High Performance Support for Lecturing and Learning

Marco Bohnsack
data-quest GmbH
Bohnsack@data-quest.de

Abstract

This article provides information about the concept designs and features of the e-learning software "Stud.IP" (open source). The functional range covers timetables, upload of documents, discussion boards, personal homepages, chat rooms etc. The goal is to offer an up to date infrastructure of both lecturing and learning. In addition, institutes get a high performance environment for the administration of their personnel, care of their web pages and the automatic generation of meeting- or personnel lists. Operators can access a reliable support system and may take part in a widespread developer community.

1. History and basic concepts of Stud.IP

1.1 History

In contrast to most e-Learning-platforms and course-management-software, Stud.IP (stands for "Studienbegleitender Internetsupport von Präsenzlehre" which means study accompanying, internet-based support of lecturing/learning) was developed without major financial subsidy. In 1999 a team of lecturers and students came together at the Centre for Media Science at the University of Göttingen. The group intended to conceive, design and program an efficient, easy-to-use, data-based internet-platform which should fit to the specific needs of both lecturers and students and their courses at university. The first version of this software was successfully tested in fall 2000 and supported all 50 courses of the Centre for Media Science. This early version already contained key elements of the actual Stud.IP, e.g. personalized accounts, a course index and the possibility to plan and organize the curriculum, forums and folders for exchanging and sharing documents. In 2002 data-quest began to coordinate the development of Stud.IP and started to offer professional support for universities, computer centers and companies on a level hitherto only known by manufacturers of closed-source-software. Including universities, colleges, companies and public services Stud.IP is currently known to be applied at over 50 institutions by more than 100.000 users in Germany, Austria, Switzerland and even in Indonesia. Despite the fact that the number of features expanded from release to release, the basic concepts did not change: one platform for the needs of administration, lecturing and learning combined with community and communication features to increase the motivation of its users.

1.2 Didactic concept

Though it is possible to integrate learning modules in the learning environment of Stud.IP, the didactic concept is not to substitute the courses by E-Learning. Courses at a german university are organized in weekly sessions. The time between two lessons should be used for repetition and preparation. Many students however realize only the lessons themselves as part of their studies. Therefore lessons can not be given as efficient as intended. Stud.IP was designed to close this gap: the subjects of the course can be discussed between the lessons, ideas can be developed together in a wiki-web, papers can be downloaded in time. The students' motivation to participate is raised by different strategies: first, high identification with Stud.IP by personalization (personal accounts, homepages within the system, community features, services like calendars, etc.). The second strategy consists in supporting unpopular duties like putting together the personal time-table, keeping the personal course record up-to-date or sharing homework with others. The second strategy also works on lecturers: the organization and administration of courses is simplified by automatic schedules, lists of participants, admission procedures, a news system and many more features.

1.3 Technical concepts

All software components that are needed by operators and providers to run an installation of Stud.IP are open source software. A classic LAMP-system is necessary, consisting of Linux, the Apache-Webserver, the MySQL-database and the scripting-language PHP. All these components offer a high performance and reliability and are downloadable for free. Stud.IP is realized in plain HTML with included dynamic content provided by the underlying databases.

Users only need a web browser of their choice and an internet connection to participate. There are no special requirements like specific browsers, plug-ins, java or additional software. Even text-based browsers like lynx can be applied. Users are also independent in their choice of operating systems - an important matter in the heterogeneous IT-environment at universities. The use of HTML and the fast working databases make Stud.IP highly performant even with extremely limited bandwidth: a 56K-Modem is wholly sufficient to access and use all Stud.IP-features.

Stud.IP also provides a couple of interfaces to import and export data in different formats, like XML or RTF. An iCal- and VCF-Interface is provided to synchronize the included Personal Information Management (PIM)-tools with other applications (e.g. outlook) or devices (eg. Smartphones, PDAs etc.). Single-Sign-On is possible while using the integrated LDAP-interface. Stud.IP provides many more interfaces to share data with other applications and data-based systems, for example interfaces for accessing library catalogues, establishing connections to ILIAS-learning-modules and the possibility to generate www-pages for institutions with dynamic content.



Figure 1: The start page of Stud.IP offers fast access to all relevant area, e.g. courses or the personal homepage

When Stud.IP was conceived design and usability played also an important role. The Graphical User Interface is consistent and intuitive to use, even for absolute beginners. It follows vaguely known user-interfaces like the Apple-OS or the result pages of the search-engine Yahoo. There is always more than one way to use a function in Stud.IP to avoid frustration of the users. A context-sensitive help-function also offers specific help for every single page within the system.

Stud.IP is designed to mirror the reality of university as a social environment. The system can reproduce the hierarchical structure of the entities of institutes, courses and persons. These parts are linked in many ways. A lecturer and his courses, for example, are always connected to an institute. The role-based system assigns every person to a role, e.g. student or lecturer. The virtual reproduction of known structures highly simplifies orientation for users.

2. Tools and features

Stud.IP provides a wide range of tools. Some features are not necessarily related to learning or lecturing but are integrated to motivate people to use the system. The main task is, however, to support lecturing and learning. The provided tools can be divided into administrative and didactic tools, although in many cases it is also possible to use them for both purposes. A didactic tool like the forum for example can also be used for administration.

2.1 Didactic tools

Every single course in Stud.IP contains the following tools:

The **forum** offers many useful features. It is structured in folders for specific subjects or lessons. Flat-view or tree-view is available to follow the threaded discussions. A search-engine for keywords, authors or subjects is integrated along to features for filtering, sorting and rating postings.

Using the **document folder** lecturers and students are able to upload and share files, e.g. papers, homework, presentations, etc. It is also possible to link copyright protected material using **automatic authentication** against secured external filesystems.

Using the **vote- and test module**, lecturers can quickly assemble achievement tests. The **course-chatroom** provides synchronous communication between sessions.

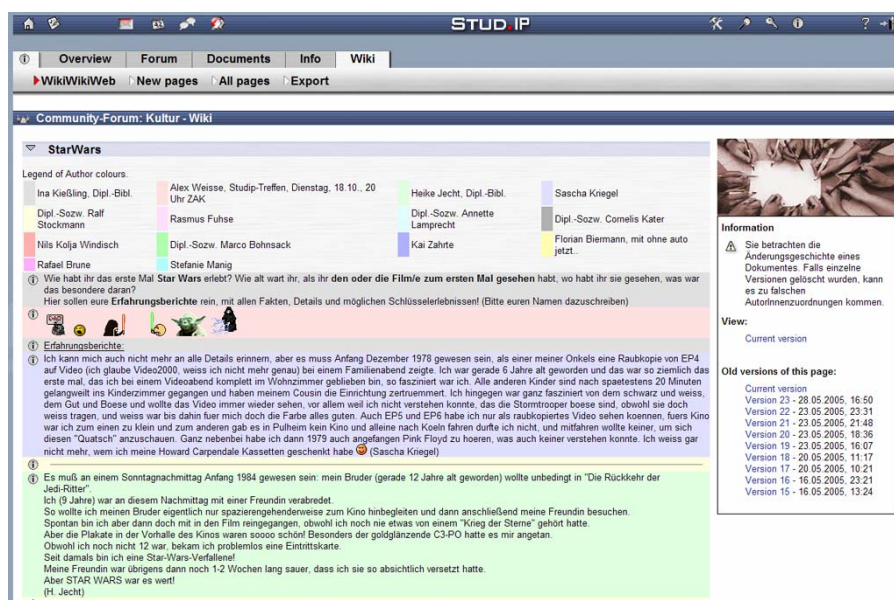


Figure 2: The wiki-web is an universal tool for collaborative work. Colors indicate different authors.

The **wiki-web** is a collaborative working environment for many different use-cases. A text can be written by several authors at the same time. Previous versions of the text are logged and stored as back-ups. It is also possible to link pages, so the wiki becomes a universal tool.

2.2 Administrative tools

The **literature-management-tool** provides direct access to search local, national or international library catalogues. Students and lectures can import or create **literature lists** and share them with other users. Export in other applications (e.g. Endnote) is possible as well.

A **smart news system** quickly distributes information, e.g. if the course room or schedule has changed. News can also be retrieved via **WAP-Interface** with every featured cell phone.

The **dynamic list of participants** with connected **group-management-tool** is an important tool for the course-administration. It can be combined with different **admission procedures** to fit the needs in courses with limited capacity. Stud.IP can generate automatically a **schedule** for the entire course including document- and forum folders for every session. The **survey-tool** can be used for either tests or evaluation of the course.

Besides the course-based tools Stud.IP offers support for the administration of institutes, too.

The integrated **resource management tool**, for example, can be used for **room-, equipment- or facility-management**. Lists of **staff, contacts and office hours** are generated dynamically and can be exported to or included in existing www-pages.

2.3 Raising motivation: communication and community features

Following the claim “users in the focus” Stud.IP is designed to provide even features for personal convenience or fun. Especially between students this leads to a strong identification with the system. In many cases people use Stud.IP not only for learning purposes. At the installation of the University of Göttingen there are, on an average, about 30 users online – even in the middle of the night. They are motivated to use the platform for communication and community purposes. This strong motivation is seldom observed in learning systems. This is achieved by simple arrangements. The score mechanism, for example, leads to much more participation. The concept is simple: for every single action performed by a user he receives score-points. Depending on the sum of the score, he is assigned a title from “greenhorn” over “expert” to “light bearer”. After the score-function was implemented into Stud.IP, the number of actions performed by users (e.g. postings in forums) increased highly.

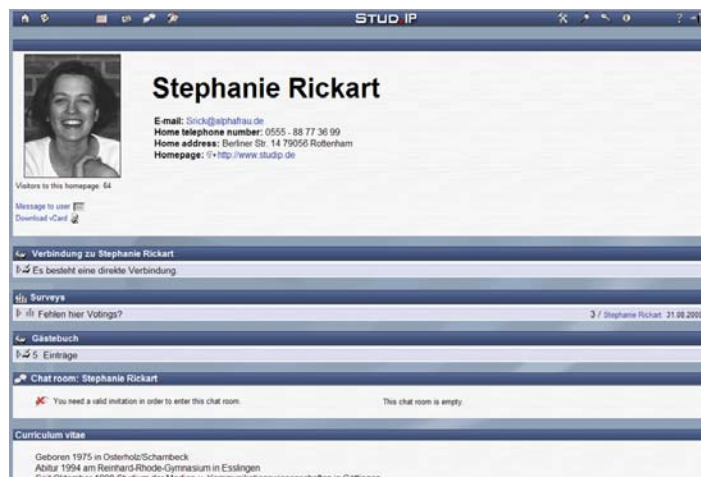


Figure 3: Personal homepage of a user with integrated features for communication.

The Stud.IP-concepts of motivation and the claim to mirror reality implicates a strong emphasis on communications. Therefore many features focus on the inter-human communication, such as the mentioned forums, wikis and chats. Combined with awareness-components (like “who-is-online” or “friend-of-a-friend”-indicators) is the internal messaging-system, which provides forwarding via external e-Mail or an instant messenger. Every user gets an internal, personal homepage, which basically contains the information given at the registration: the real name and the corresponding e-mail-address. Every user homepage offers a guest book, a chat and the possibility to send a message. Users are free to add content to this homepage, like pictures, curriculum vitae or something completely different. HTML-Tags are not allowed, but Stud.IP comes with its own tools to format texts. In all parts of Stud.IP in which names occur (e.g. at participants lists) the user’s names are directly linked to the corresponding homepage and the communication features. This offers many possibilities to contact other users and supports both learning in workgroups and knowledge-management.

3. Sustainability and development

As mentioned above, data-quest provides technical and administrative support for Stud.IP on a level unusual for open-source-software. After a security related problem is detected, data quest delivers usually within four hours critical bug fixes and security patches to customers. In addition hosting, updates, back-up-solutions, modifications, programming of new modules or changes in the design in order to follow the customers CI are offered. Besides, data-quest also offers consulting on the subjects of e-Learning, scientific and statistical research and a special version of Stud.IP for projectmanagement and –controlling.

However, data-quest is not the exclusive developer of Stud.IP. A huge, strongly connected community of free programmers, users and members of computer centres participate in the development. A board called the “CoreGroup”, consisting of 15 persons including the staff of data-quest, is monitoring ongoing development, decides about the roadmap, runs tests, checks for code and design quality and maintains releases. Two times a year a major update is released at www.sourceforge.com, www.campusource.de and www.studip.de. So sustainability and future development is guaranteed not alone by data-

quest, but also the CoreGroup, the community and some development centres at different universities e.g. Osnabrueck, Oldenburg, Halle, Trier, Rostock and Göttingen.

4. Synopsis: Features of Stud.IP

Stud.IP offers:

Support for Lecturing and courses

Forum, chat, schedule, document folders, votes and tests, evaluations, wiki-webs, literature list and uplink to library catalogues, lists of links and participants, connections to ILIAS-learningmodules, archive for expired courses, admission procedures, news, etc.

Support for learning and curriculum

Index of courses, search-engine for persons, courses, learning modules and institutes, automatic generation of Time-tables, calendar with time planer, directory for addresses, etc.

Support for Administration

Management of resources, export of content in rtf, pdf, html and xml, administration-module for mirroring complex structures of institutions, LDAP-authentication, interfaces for connections with other , etc.

Community features

Global and personal chats, homepages for every user, awareness components: "who-is-online", "friend-of-a-friend-indication", buddy lists, messaging system, free forums for discussions concerning e.g. movies, music or politics, score lists, guest books, etc.

Global features

TeX-support, role-based rights-management, simple and easy to handle GUI, WAP-Interface, RSS-Feeds, multilingual, features every browser and OS, etc.

Stud.IP-eProjectmanagement (special version distributed by data-quest)

Project management and documentation, controlling module, advanced groupware, recording and management of working-hours, CRM-functionality, etc.

This work has been carried out with the support of the European Community

The logo for 'e Learning' features a blue square with a white lowercase 'e' on the left, followed by the word 'Learning' in a black serif font.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 2.5
Netherlands License: <http://creativecommons.org/licenses/by-nc-nd/2.5/nl/>

ISBN 90 358 2318 4
NUR 840