Open Source and Open Standards

CHAPTER 31: OPEN SOURCE AND OPEN STANDARDS

Rob Koper

Open University of the Netherlands

rob.koper@ou.nl

31.1 INTRODUCTION

The objective of this chapter is to create an understanding of the importance of open source software and open standards (OSS/OS) for e-learning research. Open source is a fundamental new way to develop software, and open standards are needed to make software components work together. Both stimulate exchange, collaboration, interoperability and convergence of knowledge and these are beneficial requirements for future e-learning research.

E-learning can be defined as the use of information and communication technologies (ICTs) to facilitate and enhance learning and teaching. E-learning research is aimed at the development of new technologies to improve learning, training and teaching in various ways:

- by making it more *accessible* to everyone at any place and at any time;
- by making it more *effective* by facilitating the implementation of advanced pedagogical and organizational approaches;

- by making it more *efficient* by providing advanced (partly automated) support mechanisms for learners and teachers to perform their various tasks;
- by making it more *attractive* to users by providing adapted tasks and resources.

E-learning research is *technology oriented* instead of *theory oriented*. Technology oriented research, also called technology development or engineering, differs in fundamental ways from theory oriented research. These research approaches differ in the ways in which (a) problems are addressed, (b) research activities are performed, (c) notation and communication means that are used, and (d) the results that are delivered (see, for example, Gibbons, 2000; Hannay & McGinn, 1980; McGinn, 1978; Rogers, 1995; Simon, 1969; Vincenti, 1990). Mitcham (1994) states that: "Virtually all historians … use the word 'technology' to refer to both ancient and modern, primitive and advanced making activities, or knowledge of how to make and use artifacts, or the artifacts themselves" (p.116). A distinction can be made between (a) the technological *activities* of the researchers (methods for making an artifact), (b) the technological *knowledge* that is a result of these making activities (models and specifications) and (c) the technological *artifacts* that are the results of these activities. These distinctions will be used to structure this chapter when we discuss the use of OSS/OS in e-learning research. The following questions will be answered:

a) How does OSS/OS facilitate the technological *activities* of the researchers in terms of methodology, collaboration and dissemination of results?
b) How does OSS/OS facilitate the development of technological *knowledge* in the field?
c) How does OSS/OS facilitate the development of technological *artifacts* in the field?

Before going into these questions, the concepts of open source software (OSS) and open standards (OS) are discussed more in general with emphasis on the use of OSS/OS as means to perform research on e-learning.

31.2 OPEN SOURCE

31.2.1 What is open source?

Software is written in a computer language before it is compiled into binary code that computers can run. The human readable text originally written by the programmers in a computer language is called the source code of a program. The source code, the derived binary code, and the documentation are protected by intellectual property rights (IPR). Only the owner of the IPR is entitled to change the code or the documentation, and only the owner of the copyright is entitled to copy and distribute these.

This closed source software approach has been under attack by the free software and open source movement. The free software foundation (FSF), founded by Richard Stallman in 1984, is the organization behind MIT's GNU project. One of

the contributions of the FSF is the development of the General Public License (GPL, 2006) to protect the IPR of contributers and prevent unwanted commercialization of the software.

The FSF also maintains the free software definition: “the freedom of all users to run, copy, distribute, study, change and improve software. Source code is seen as a kind of scientific knowledge that should be published to facilitate innovation” (see FSF-DEF, 2006). It is worth noting that free software as used here does not coincide with the notion of software that is available without cost to the user. Some software that is freely available is not free software and some free software may involve nominal costs to users.

The term ‘free software’ has developed some negative connotations, especially in industry. This is the reason why a group of people, including Eric Steven Raymond, started in 1997 to promote the use of free software by stressing the technical superiority and low cost instead of its rather anti-business and ideological aspects. They use the term ‘open source software’ instead of ‘free software’ and founded the Open Source Initiative (OSI) in 1998 to provide a definition of OSS and a set of criteria for open source licenses (OSI-licenses, 2006). The GPL license is considered to be a valid OSI license, among many other licenses that are less restrictive for use in the commercial world.

31.2.2 Open source development model

There are now many OSS development projects. SourceForge®, for example, supports more then a 100,000 projects, most for general use but many specifically developed for e-learning. One of the characteristics of OSS is that it is developed in a different way than commercial software. Raymond (1998, 2001) compared two development models: the cathedral (as a metaphor for traditional software development) and the bazaar (as a metaphor for OSS development) to ground the idea of higher quality and lower costs. Characteristics of the bazaar model for OSS development tend to include: (a) globally distributed communities of developers collaborating primarily through the Internet, (b) developers working in parallel, (c) developers exploiting the power of peer review for debugging and requirements analysis, (d) rapid, incremental release schedules, and (e) projects with pools of experienced and esteemed professional developers (Feller & Fitzgerald, 2002). OSS communities have developed some strong cultural norms that govern the mainly self-organized development system (Bergquist & Ljungberg, 2001; Jorgensen, 2001).

The success of the OSS development model has invoked many questions, especially in economics and organizational theory. Madey, Freeh and Tynan (2002, p.1807) formulate it this way:

> "The OSS movement is a phenomenon that challenges many traditional theories in economics, software engineering, business strategy, and IT management. Thousands of software programmers are spending

> tremendous amounts of time and effort writing and debugging software, most often with no direct monetary compensation“

Empirical studies have been performed on the size and distribution of development teams (Crowston & Howison, 2005), comparisons of the organization of different OSS projects (Dempsey et al., 1999; Gallivan, 2001; Mockus, Fielding & Herbsleb, 2002), the organization of social relationships and incentives in OSS communities (Bergquist & Ljungberg, 2001; Lerner & Triole, 2002).

Although an OSS development model is sometimes perceived as something new, the principle of sharing software and co-developing has been a common practice among academics from the early days of computer programming. In the early 1960s many fundamental software programs (operating systems, computer languages, etc.) were developed in universities such as Massachusetts Institute of Technology and The University of California at Berkeley, and in company laboratories like AT&T Bell Labs and Xerox’s Palo Alto Research Center. Researchers shared their code for others to inspect, to use and to improve. This mode of working was rather similar to the way in which researchers have always shared ideas through publications, reports, notes and conferences. This mode of sharing was rather informal and as a result of this AT&T could start to enforce its IPR on UNIX in the beginning of the 1980s. This has been one of the triggers for the development of the GNU license.

31.2.3 Some general open source applications

Many of the tools and services used in daily life are based on OSS, specifically on the server side (for example, email based on *sendmail*, Websites based on *Apache*, servers running on *Linux*). For the client there are also many high quality OSS alternatives for commercial software available. For example: Thunderbird (2006) as a mail client, Firefox (2006) as a browser, the GIMP (2006) as an advanced drawing tool, Freemind (2006) as a mind mapping tool, ECLIPSE (2006) as an integrated software development environment, or OpenOffice.org (2006) as an Office Suite. Most of these tools are interoperable with commercial software through the use of import and export filters to different formats.

31.2.4 Open source in learning, education and training

OSS is used in a variety of ways in learning, education and training contexts. Many types of computer use involve some kind of informal learning, like performing a search with Google, using Wikipedia, making and using podcasts, writing and reading Blogs and Wiki's, and so on.

The use of OSS in schools is explored and applied in many regions in the world where cost savings and stimulating local industries are important issues. In these cases, the Linux operating system is often used as a base for an educational software package that contains a selection of general and specific open source applications that can be used in the schools (see, for instance, Edubuntu, 2006). In the Spanish region Extremadura, a Debian-based version of Linux (LinEx, 2006),

is deployed on some 70,000 desktop PCs and 400 servers in the educational sector. In Norway around 200 schools use Skolelinux (2006).

Besides these products, there are many projects that are delivering specific educational applications. In the area of e-learning, there are several open source learning management systems (LMSs) including Moodle (2006), Sakai (2006), DotLearn (2006), Bodington (2006), aTutor (2006), Dokeos (2006), and many more systems.

The development of these LMSs is at the moment challenged by the emergence of new generations of technologies. For instance:

- the use of Web services for e-learning (Alonso, Casati, Kuno, & Machiraju, 2004; Vossen & Westerkamp, 2003).
- the use of semantic Web principles (Anderson & Whitelock, 2004; Berners-Lee, Hendler, & Lassila, 2001),
- the use of adaptive learning principles (Berlanga & Garcia, 2005; Brusilovsky, 2001; De Bra, Aroyo, & Chepegin, 2004),
- the use of learning process oriented systems (Dalziel, 2003; LAMS, 2006; Paquette et al., 2005),
- the use of social software (ELGG, 2006),
- the use of shared, self-created multimedia files like podcasts (Lionshare, 2006), and

- the use of mobile technologies in learning (Jones, Kukulska-Hulme, & Mwanza, 2005).

BECTA (2005) has performed a study to the use of OSS in UK schools. The cases presented in the study show that OSS can be used as: server operating system, desktop operating system and for applications used in the classroom or for administration. The study also mentions the reasons why schools moved to OSS:

- they liked its transparency and flexibility, which made it possible to alter the software according to their needs;
- there was an educational value to providing pupils with a broader experience of operating systems and software;
- it was a way to achieve value for money and to extend the ICT network and facilities;
- they had access to appropriate knowledge, skills and experience to support an OSS implementation;
- Most stakeholders (pupils, teachers, parents) also appreciated the use of OSS.

The disadvantages identified were: lack of curriculum specific courseware, compatibility problems with some commercial software, lack of familiarity among teachers and pupils. The study has also found that the total costs for the use of OSS were lower than the use of proprietary software, but this is dependent largely on the way OSS is used and supported.

31.3 OPEN STANDARDS

31.3.1 What are Open Standards?

Open standards are of enormous importance in our society to ensure that products and services are of sufficient quality and can work together, that is to say, are interoperable.

The term ‘open standard’ has many interpretations. According to Krechmer (2005) this is due to the fact that creators, implementers and users of OS each have a different set of requirements and as a result a different perspective on OS. For instance, the creators – as represented by the standards organizations - will focus on the openness of the process of standard development, specifically a due process with open meetings and decisions made by consensus. The implementers will focus more on the free use of the standards and the compatibility with previous implementations. The end-users will focus on aspects like the number of implementations from different vendors and the compatibility with currently used systems. End-users are often interested in de facto standards instead of OS, for instance when they say that they want to ‘standardize’ on Blackboard.

The definition that I will use for OS (in e-learning) is: Open standards (in e-learning) are commonly agreed upon and published specifications of the conventions used in a community to ensure the quality and/or interoperability of (e-learning) products and services.

Several remarks can be made about this definition. First of all the definition contains the word ‘specification’, which means that an open standard is conceived as a document or set of documents and not as a specific product or service. These documents contain agreements about quality standards, data formats and/or communication protocols.

The idea is that e-learning products implement the different open standards to enable interoperability. Many products also advertise that they are compliant with certain standards, but this is hard to test because there is still a lack of formal conformance procedures to test whether a product is truly compliant with a specific standard. The European TELCERT (2006) project has developed a first set of tools for such conformance testing.

Second, the definition defines two core functions of OS:

1. To ensure the *quality* of e-learning products and services, including the quality of learning objects, the quality of a systems design, the usability of the software, and so on.
2. To ensure the *interoperability* of e-learning products and services. Interoperability supports the collaboration between systems, but also between humans who develop or use a system (for example, notation standards and standard vocabularies).

Furthermore, the definition points to the fact that standards are always agreed-upon and used *within a certain community* of interest, such as a company, a consortium, a country, a specific technology, or a (worldwide) field of expertise.

A standard has the status of a recommendation for the community members. This leaves open the possibility that there are different communities and different standards in the same area, like there are different power plug standards or railway systems in different countries. For metadata we can use the IMS/IEEE LOM (educational sector) or the Dublin Core Metadata (library sector). Added to this, there are many countries, professional sectors and companies that have defined their own local standards that are not compatible with each other, nor with international standards.

The ideal of the standards committees is to have one world-wide accepted standard that can be localized to fit the needs of different organizations. This ideal is, however, hard to accomplish in practice, so we have to deal with the fact that there are still many different communities, each with their own incompatible standards. In that case the interoperability question has to be solved when the two communities want to collaborate.

31.3.2 How are open standards developed and which e-learning standards are available?

International specifications are traditionally developed by three standards organizations: the International Organization or Standardization (ISO, founded in 1947), the International Electrotechnical Commission (IEC, founded in 1906) and the International Telecommunication Union (ITU, founded in 1865). Most countries and parts of the world have their own standards organizations that are

directly associated with the international organizations (for example, ANSI in the USA, CEN in Europe).

Besides these organizations that are country or regionally specific, there are also standards organizations that transcend country boundaries by adopting an expert model approach, like the Institute of Electrical and Electronics Engineers (IEEE). Some well known IEEE standards are the IEEE 802.3 Ethernet standard and the IEEE 802.11 Wireless Networking standard. The IEEE is also active in e-learning through the IEEE Learning Technologies Standards Committee (LTSC). This committee is working on topics such as a digital rights expression language, computer managed instruction, learning objects metadata, and competency definitions.

ISO formed a joint technical committee with the IEC (ISO/IEC JTC1) that has the objective to develop, maintain, promote and facilitate ICT standards required by global markets meeting business and user requirements concerning: design and development of systems and tools; performance and quality of products and systems; security of systems and information; portability of application programs; interoperability of products and systems; unified tools and environments; harmonized vocabulary and user friendly and ergonomically designed user interfaces.

JTC1 has several sub-committees including SC36, which is responsible for Information Technology for Learning, Education and Training. SC36 is working on standards for a Collaborative Workplace, for Agent/Agent communication and

on a Learner to Learner Interaction Scheme. They are currently setting up many more groups.

The standards organizations that work according to the countries model, specifically ISO, have a reputation in the 1990s of being too slow to keep up with the standardization needs of fast changing areas like ICT and e-learning. ISO had a large project, Open Systems Connect, that tried to develop a common computer networking standard. The project did not succeed and was stopped in 1996 and in fact it was run over an organization called the Internet Engineering Task Force (IETF, founded in 1986). This task force had a less bureaucratic, open process and developed the basic protocols suits that were needed for the Internet to operate. Later, the IETF was also perceived as too slow and most industry vendors are currently working with specifications from more specialized consortia like the World Wide Web Consortium (W3C). This consortium creates and maintains standards for the World Wide Web (HTTP, URL, Linking, XML, Semantic Web). In the e-learning field the dominant specialized consortium is IMS, a consortium of the major players, companies and researchers, in the e-learning field. IMS developed and maintains 17 specifications in the following fields: metadata, assessment, learning design, content packaging, sequencing, ePortfolio, learner information, digital repositories, competency definition and interoperability of learning management systems with enterprise systems.

Most of the current e-learning standards concentrate on the syntax of the data format that should be used for the asynchronous exchange of learning resources or

learner information. Less attention has been given to the standardization of the synchronous communication between systems and the standardization of the semantics of the communication process. In the field of semantics there are several exceptions, for instance the work done by the International Board of Standards for Training, Performance and Instruction (IBSTPI; see http://www.ibstpi.org) on the definition of competencies for instructors, instructional design, training managers, and evaluators. However, these standards are not (yet) defined in the technical formats provided by IMS, IEEE or HR-XML.

31.4 OSS/OS AS A MEANS TO FACILITATE E-LEARNING RESEARCH

How does or can OSS/OS facilitate e-learning research? As stated before, OSS/OS can facilitate e-learning research that results in both new technological knowledge and new technological artifacts. In the next paragraphs I will discuss these possibilities of using an OSS/OS approach in e-learning research.

31.4.1 OSS/OS to facilitate technological activity in e-learning

The major activity in e-learning research is to *develop* new e-learning technologies. This development process is facilitated by research methodologies based on the principles of systems engineering (see also Richey & Nelson, 1996). The use of OSS/OS can facilitate the development process in several ways: a) by providing a standard notation system to foster communication and collaboration, b) by facilitating the development of the systems by multiple distributed users

using the OSS development model; c) by facilitating the evaluation of the developed artifacts, and d) by stimulating the dissemination of results.

During this development there is a need to communicate the design of the system among researchers and users within and outside the team. For communication purposes, a *notation system* is used to capture user requirements and to notate the design of the envisaged system. When such a notation system conforms to a widely known open standard, it facilitates the correct understanding of the design and so the quality of the discussions among developers and users. The best example of such a notation system is the Unified Modeling Language (UML) (Booch, Rumbaugh, & Jacobson, 1999; Fowler, 2000), an open standard developed by the Object Management Group (OMG). The use of UML is nowadays very common in ICT research, but still a rather new phenomenon in e-learning research. However, also in e-learning it is used more and more in publications (see Zarraonandia, Dodero, & Fernández, 2005).

UML defines nine types of diagrams each providing a different view on a system under development. Three types of diagrams in e-learning publications are most often used:

1. Use Cases to model the (envisaged) user requirements and benefits of the system for (future) users (see Figure 2 of Asensio et al., 2004).
2. Class Diagrams are used to model the core entities (concepts or classes) and their relationships in the problem domain. These diagrams

can also be used to express a domain ontology or to design the data structures in an application (see Figure 2 of Koch & Wirsing, 2002).

3. Activity Diagrams are used to model the processes or workflow in a system (see Figure 3 of Derntl & Motschnig-Pitrik, 2003).

The UML diagrams are shared among researchers in publications and they support collaboration and communication during the process of analysis and design. They are also used in group communication to create a conceptual model that integrates different perspectives. An example is the UML that was developed to integrate the classical and modern views on assessment (Joosten-Ten Brinke et al., 2005).

Based on the UML models, OSS can be developed implementing OS. OSS/OS can be very beneficial to the project: new systems can be built by re-using OSS code or by adapting the code of existing systems. The use of OS makes the inclusion of existing services or data possible.

Another advantage of using OSS/OS is that it can facilitate the evaluation of the system in various ways. First of all the advocated Bazaar methods of OSS can have advantages in the quality of the code. To use the terms of Raymond (2001): release early and often, involve the users, many eyeballs tame complexity, and given enough eyeballs, all bugs are shallow. Furthermore it can have advantages in the set-up of the experiments itself. When the software is made available and a user base is developing, these same users can be used for the evaluation research.

This type of research can aim for high ecological validity (see also Gilbert & Troitzsch, 1999).

Another advantage of using OSS/OS in e-learning research is easy dissemination through channels like SourceForge®. This provides a natural means to attract new initiatives, to improve the software, to use the software when it is at a certain quality, and so on. A basic requirement is that the product addresses a need perceived by a community of users.

31.4.2 OSS/OS to facilitate the development of Technological Knowledge

One of the results of technological activities is the development of new technological knowledge. Technological knowledge is knowledge that describes how an artifact (or a system) can be made and how this artifact can be used. This knowledge is captured in:

a.the UML diagrams that are used to design the system,

b.the code of the system that has been developed,

c.the documentation of the system that provides a user perspective of the system,

d.the publications about the evaluation of the system.

When using an OSS approach this knowledge is available for all researchers to test, replicate or elaborate. In closed source systems or in regular research approaches the access to the design and the code of the system is restricted, so the knowledge is only partially available. The OSS approach can lead to more convergence of knowledge in the field when researchers adopt the habit of using

and adapting what is already available, instead of building everything again from scratch.

The use and development of OS has another advantage in research. OS can be seen as consolidated, agreed upon knowledge about the data structure, functionality or the semantics of a system. Standards commissions provide a platform to converge divergent theories and models to the best possible abstraction of the current state-of-the-art. For instance the IMS QTI specification summarizes the set of different test items that are frequently used in education. This process itself is a strong means in the field to come to an agreement and summary of some aspects in the field. Especially when researchers have the habit to contribute to the standards committees, use the standards that are released as much as possible, and test the specifications to identify strong and weak points.

31.4.3 OSS/OS to facilitate the development of technological artifacts

Besides the creation of knowledge, the core result of technological activity is the development of the technological artifacts. In e-learning research we produce the following types of artifacts: models and open standards, software and documentation. In the previous paragraphs many examples of the OSS/OS artifacts were mentioned, for example: (a) specifications such as IMS QTI, IEEE LOM, and (b) software such as Moodle, Lionshare, Sakai.

31.5 EXAMPLE

The development and use of the open standard IMS Learning Design (IMSLD, 2003; Koper & Olivier, 2004; Koper & Tattersall, 2005) and the related OSS will demonstrate the use of OSS/OS in e-learning research.

The IMS LD model (technological knowledge)

IMS Learning Design (LD) is an open standard that is used to specify the design of a teaching and learning process in a machine interpretable way. LD can be seen as a formal instructional design language. The specification consists of a set of documents and an XML Schema Definition that supports the coding of the learning design of courses in XML format. At the base of the specification is the Conceptual Model as presented in Figure 31.1.

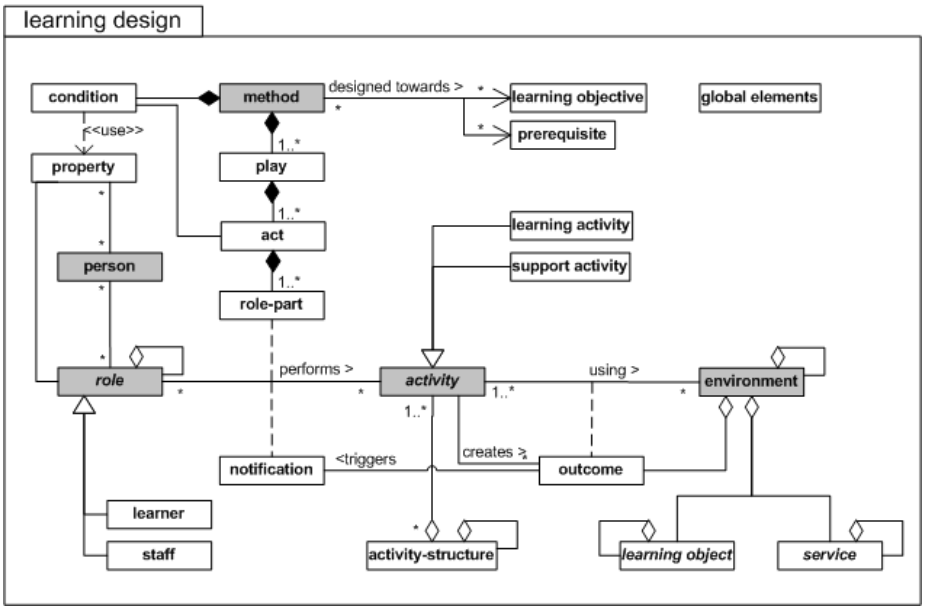

Figure 31.1. The Conceptual Domain Model of IMS Learning Design in UML

This model is specified as a UML class model and is slightly adapted from the EML model that has been developed by studying and abstracting different instructional design approaches (Koper & Manderveld, 2004). It was tested, discussed and adapted in different situations using different technologies (Tattersall, Vogten, & Hermans, 2005; Van Es & Koper, 2005), and this process still goes on, although a first stable release of the open specification was reached after five years of work (1998-2002). The model itself can be seen as a technological theory that describes a large variety of different instructional design approaches. In this way it serves as a convergence mechanism in the field of instructional design. In essence it should be possible to describe and implement most of the current instructional design models (Reigeluth, 1999) in LD. The fact that this is an open specification has invoked many different research initiatives in the area of ontologies for learning design, patterns in learning design, runtime adaptations and the design and development of a variety of authoring and runtime tools (for an overview see Koper, 2005).

Open source software for IMS LD

The availability of LD as an open standard has stimulated the development of many tools to support the specification. Most of these tools are developed as OSS in academic settings, for example, as part of PhD research. Griffiths and colleagues (2005) provide an overview of available tools and their classification. Major research issues are the development of more user-friendly and integrated authoring and runtime tools (Hernández-Leo, Asensio-Pérez, & Dimitriadis,

2006), Authoring of adaptive learning designs (Van Rosmalen et al., 2005), the graphical representation of learning designs (Paquette et al., 2006), the integration of assessment in learning designs (Joosten-ten Brinke et al., 2005; Pacurar, Trigano, & Alupoaie, 2004) and the use of semantic web tools with learning design (Amorim et al., 2006; Knight, Gašević, & Richards, 2006).

31.6 CONCLUSION AND SOME PRACTICAL TIPS

The concepts of OSS and OS have been elaborated in this chapter. The discussion suggests that OSS and OS can improve the convergence of knowledge in the field, improve the general quality and interoperability of e-learning applications, and improve collaboration between researchers and users. How might one use OSS and OS in research? As a conclusion I will try to summarize this:

- First it is important to learn how to read and create UML diagrams to specify a system that solves some real problems in the field of learning and/or teaching.
- Furthermore it is important to study the existing open standards in the e-learning field, specifically the ones from IMS and IEEE (and ISO when they are becoming available). You will need some knowledge about XML (2003), RDF (2003) and standards organizations. Use the references in this chapter.

- Download, install, test and use a variety of open source programs and distributions. To not disturb your daily work on your computer you can install virtual machine software like VMware for experimentation. This will enable you to install different Linux distributions (for example, Edubuntu).
- Optional: learn to code and participate in an open source development project or start one on your own.
- Use existing OSS as much as possible in your work and be strict in the use of OS were possible. This stimulates convergence and collaboration.
- Participate in relevant communities: demonstrate your work in workshops, write conference papers and journal papers about your work. Provide relevant feedback and input to standards committees.

REFERENCES

Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2004). *Web services: Concepts, architectures and applications*. Heidelberg: Springer.

Amorim, R. R., Lama, M., Sánchez, E., Riera, A., & Vila, X. A. (2006). A learning design ontology based on the IMS specification. *Educational Technology & Society, 9*(1), 38-57.

Anderson, T., & Whitelock, D. (2004). The educational semantic Web: Visioning and practicing the future of education [Electronic Version]. *Journal of Interactive Media in Education*, 1. Retrieved August 22, 2006, from http://www-jime.open.ac.uk/2004/1/editorial-2004-1-disc-paper.html#EditorialStart.

Asensio, J. I., Dimitriadis, Y. A., Heredia, M., Martinez, A., Alvarez, F. J., Blasco, M. T., et al. (2004, February 11-13). *Collaborative learning patterns: Assisting the development of component-based CSCL applications*. Paper presented at the 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing, Coruña, Spain.

aTutor. (2006). aTutor: open source learning management system. Retrieved August 22, 2006, from http://www.atutor.ca/

BECTA. (2005). *Open source software in schools*. Retrieved August 22, 2006, from http://www.becta.org.uk/corporate/publications/documents/BEC5606_Full_report18.pdf.

Bergquist, M., & Ljungberg, J. (2001). The power of gifts: Organising social relationships in open source communities. *Information Systems Journal, 11*(4), 305-320.

Berlanga, A., & Garcia, F. J. (2005). Learning technology specifications: semantic objects for adaptive learning environments. *International Journal of Learning Technology, 1*(4), 458-472.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American* (May 17).

Bodington. (2006). Bodington: Open source learning management system. Retrieved August 22, 2006, from http://bodington.org/

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language user guide*. Reading: Addison-Wesley.

Brusilovsky, P. (2001). Adaptive hypermedia. *User modeling and user adapted interaction, 11*(1/2), 87-110.

Crowston, K., & Howison, J. (2005). The social structure of free and open source software development [Electronic Version]. *First Monday*, 10. Retrieved August 22, 2006 from http://firstmonday.org/issues/issue10_2/crowston/index.html

Dalziel, J. (2003, December 7-13). *Implementing Learning Design: The Learning Activity Management System.* Paper presented at the Interact, Integrate,

Impact Conference: Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education, Adelaide.

De Bra, P., Aroyo, L., & Chepegin, V. (2004). The next big thing: Adaptive Web-based systems [Electronic Version]. *Journal of Digital Information*, 5. Retrieved August 22, 2006 from http://jodi.tamu.edu/Articles/v05/i01/DeBra/

Dempsey, B., Weiss, D., Jones, P., & Greenberg, J. (1999). A quantitative profile of a community of open source linux developers [Electronic Version] from http://www.ibiblio.org/osrt/develpro.html

Derntl, M., & Motschnig-Pitrik, R. (2003). *Conceptual Modeling of Reusable Learning Scenarios for Person-Centered e-Learning.* Paper presented at the International Workshop Interactive Computer aided Learning, Villach.

Dokeos. (2006). Dokeos: open source learning management system. Retrieved August 22, 2006, from http://www.dokeos.com/

DotLearn. (2006). LRN: open source learning management system. Retrieved August 22, 2006, from http://www.dotlrn.org/

ECLIPSE. (2006). Open source development platform and application frameworks for building software. Retrieved August 22, 2006, from http://www.eclipse.org/

Edubuntu. (2006). Edubuntu: linux for young human beings. Retrieved August 22, 2006, from http://www.edubuntu.org/

ELGG. (2006). Open source 'learning landscape'. Retrieved August 22, 2006, from http://elgg.net

Feller, J., & Fitzgerald, B. (2002). *Understanding open source software development*. London: Addison-Wesley.

Firefox. (2006). Firefox open source web browser. Retrieved August 22, 2006, from http://www.mozilla.com/firefox/

Fowler, M. (2000). *UML distilled* (2nd ed.). Upper Saddle River, NJ: Addison-Wesley.

Freemind. (2006). Free mind mapping software. Retrieved August 22, 2006, from http://freemind.sourceforge.net/wiki/index.php/Main_Page

FSF-DEF. (2006). Free Software Foundation [website of the association]. Retrieved August 22, 2006, from http://www.fsf.org/licensing/essays/free-sw.html

Gallivan, M. J. (2001). Striking a balance between trust and control in a virtual organization: a content analysis of open source software case studies. *Information Systems Journal, 11*(4), 277-304.

Gibbons, A. (2000). *The Practice of Instructional Technology*. Presented at the 2000 Association for Educational Communications and Technology conference, Longbeach (CA). Retrieved August 22, 2006, from http://www.aect.org/intranet/publications/index.asp

Gilbert, N., & Troitzsch, K. G. (1999). *Simulation for the social scientist*. Buckingham: Open University Press.

GIMP. (2006). GNU image manipulation program. Retrieved August 22, 2006, from http://www.gimp.org/

GPL. (2006). General public license. Retrieved August 22, 2006, from http://www.gnu.org/copyleft/gpl.html

Griffiths, D., Blat, J., Garcia, R., Vogten, H., & Kwong, K. (2005). Learning design tools. In R. Koper & C. Tattersall (Eds.), *Learning design: A handbook on modelling and delivering networked education and training* (pp. 109-136). Berlin-Heidelberg: Springer Verlag.

Hannay, N.B., & McGinn, R.E. (1980). Anatomy of modern technology: Prolegomenon to an improved public policy for the social management of technology. *Daedalus, 109*(1), 25-53.

Hernández-Leo, D., Asensio-Pérez, J. I., & Dimitriadis, Y. (2006). Computational representation of collaborative learning flow patterns using IMS Learning Design. *Educational Technology & Society, 8*(4), 75-89.

IMSLD. (2003). IMS Learning Design Specification. Retrieved August 22, 2006, from http://www.imsglobal.org/learningdesign/index.cfm

Jones, A., Kukulska-Hulme, A., & Mwanza, D. (2005). Portable learning - Experiences with mobile devices [Electronic Version]. *Journal of Interactive Media in Education*. Retrieved August 22, 2006 from http://www-jime.open.ac.uk/2005/21/

Joosten-Ten Brinke, D., Van Bruggen, J., Hermans, H., Latour, I., & Koper, R. (2005). *Conceptual model for assessment.* Paper presented at the European Association for Research on Learning and Instruction, Nicosia, Cyprus.

Jorgensen, N. (2001). Putting it all in the trunk: Incremental software development in the FreeBSD Open Source Project. *Information Systems Journal, 11*(4), 321-336.

Knight, C., Gašević, D., & Richards, G. (2006). An ontology-based framework for bridging learning design and learning content. *Educational Technology & Society, 9*(1), 23-37.

Koch, N., & Wirsing, M. (2002). *The Munich reference model for adaptive hypermedia applications.* Paper presented at the Adaptive Hypermedia and Adaptive Web Based Systems, LNCS 2347, Malaga.

Koper, E. J. R. (2005). Current research in learning design. *Educational Technology & Society, 9*(1), 13-22.

Koper, E. J. R., & Manderveld, J. M. (2004). Educational modelling language: Modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology, 35*(5), 537-552.

Koper, E. J. R., & Olivier, B. (2004). Representing the learning design of units of learning. *Educational Technology & Society, 7*(3), 97-111.

Koper, E. J. R., & Tattersall, C. (2005). *Learning design: A handbook on modelling and delivering networked education and training*. Berlin-Heidelberg: Springer Verlag.

Krechmer, K. (2005, January). *The meaning of open standards.* Paper presented at the 38th Annual Hawaii International Conference on System Sciences, City, Hawaii.

LAMS. (2006). LAMS: open source activity management system. Retrieved August 22, 2006, from http://www.lamsfoundation.org

Lerner, J., & Triole, J. (2002). Some simple economics of open source. *Journal of Industrial Economics, 50*(2), 197-234.

LiNex. (2006). Linux version developed and used in the Spanish region Extremadura. Retrieved August 22, 2006, from http://www.linex.org/

Lionshare. (2006). Lionshare: Secure peer-to-peer environment for the sharing of educational resources. Retrieved August 22, 2006, from http://lionshare.its.psu.edu/main/

Madey, G., Freeh, V., & Tynan, R. (2002). *The open source software development phenomenon: An analysis based on social software theory.* Paper presented at the Eighth Americas Conference on Information Systems, Dallas, TX.

McGinn, R. E. (1978). What is technology? *Research in Philosophy and Technology* (1), 179-197.

Mitcham, C. (1994). *Thinking through technology: The path between engineering and philosophy*. Chicago: The University of Chicago Press.

Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology, 11*(3), 309-346.

Moodle. (2006). Moodle: open source learning management system. Retrieved August 22, 2006, from http://moodle.org

Openoffice.org (2006). Open source office package. Retrieved August 22, from http://openoffice.org

OSI-licenses. (2006). Open source licenses. Retrieved August 22, 2006, from http://www.opensource.org/licenses/index.php

Pacurar, E. G., Trigano, P., & Alupoaie, S. (2004). A QTI editor integrated into the netUniversité web portal using IMS LD [Electronic Version]. *Journal of Interactive Media in Education*, 9. Retrieved August 22, 2006 from http://www-jime.open.ac.uk/2004/1/editorial-2004-1-disc-paper.html#EditorialStart.

Paquette, G., De la Teja, I., Léonard, M., Lundgren-Gayrol, K., & Marino, O. (2005). An instructional engineering model and tool for the design of units of learning. In R. Koper & C. Tattersall (Eds.), *Learning design: A handbook on modelling and delivering networked education and training* (pp. 161-184). Berlin-Heidelberg: Springer Verlag.

Paquette, G., Léonard, M., Lundgren-Cayrol, K., Mihaila, S., & Gareau, D. (2006). Learning design based on graphical knowledge-modelling. *Educational Technology & Society, 9*(1), 97-112.

Raymond, E. S. (1998). The cathedral and the bazaar [Electronic Version]. *First Monday*, 3, 1-33. Retrieved August 22, 2006 from http://www.firstmonday.org/issues/issue3_3/raymond/index.html.

Raymond, E. S. (2001). *The cathedral and the bazaar*. Sebastopol, CA: O'Reilly.

RDF. (2003). Resource Description Format (RDF). Retrieved August 22, 2006, from http://www.w3c.org/RDF/

Reigeluth, C. E. (1999). *Instructional-design theories and models: Subtitle goes here (volume II)*. London: Lawrence Erlbaum Associates.

Richey, R. C., & Nelson, W. A. (1996). Developmental research. In D. H. Jonassen (Ed.), *Educational communications and technology* (pp. 1213-1245). New York: Macmillan.

Rogers, E.M. (1995). Diffusion of innovations. New York: The Free Press.

Sakai. (2006). Sakai: open source learning management system. Retrieved August 22, 2006, from http://sakaiproject.org/

Simon, H. A. (1969). The sciences of the artificial, (2nd ed.). Cambridge, MA: MIT Press

Skolelinux. (2006). Skolelinux: open source linux for education. Retrieved August 22, 2006, from http://www.skolelinux.org

Tattersall, C., Vogten, H., & Hermans, H. (2005). The Edubox Learning Design Player. In R. Koper & C. Tattersall (Eds.), *Learning design: A handbook on modelling and delivering networked education and training* (pp. 303-310). Berlin-Heidelberg: Springer Verlag.

TELCERT. (2006). TELCERT: EU R&D project aimed at conformance testing of open learning technology standards. Retrieved August 22, 2006, from http://www.opengroup.org/telcert/

Thunderbird. (2006). Thunderbird Open Source Mail Client. Retrieved August 22, 2006, from http://www.mozilla.com/thunderbird/

Van Es, R., & Koper, E. J. R. (2005). Testing the pedagogical expressiveness of LD. *Educational Technology & Society, 9*(1), 229-249.

Van Rosmalen, P., Brouns, F. M. R., Tattersall, C., Vogten, H., van Bruggen, J., & Sloep, P. B. (2005). Towards an open framework for adaptive, agent-supported e-learning. *International Journal of Continuing Engineering Education and Life-Long Learning, 15*(3-6), 261 – 275.

Vincenti, W. G. (1990). What Engineers Know and How They Know It: Analytical Studies From Aeronautical History. Baltimore: Johns Hopkins University Press.

Vossen, G., & Westerkamp, P. (2003). *E-learning as a Web service.* Paper presented at the Seventh International Database Engineering and Applications Symposium, Hong Kong.

XML. (2003). Extensible Markup Language (XML). Retrieved August 22, 2006, from http://www.w3c.org/XML/

Zarraonandia, T., Dodero, J. M., & Fernández, C. (2005). Crosscutting runtime adaptations of LD execution. *Educational Technology & Society, 9*(1), 123-137.

ACKNOWLEDGMENT

This chapter is sponsored by the TENCompetence Integrated Project that is funded by the European Commission's 6th Framework Programme, priority IST/Technology Enhanced Learning. Contract 027087 (www.tencompetence.org).