Applying Open-source solutions in E-learning environment

Darko Etinger  
Juraj Dobrila University of Pula  
Department of Economics and Tourism „Dr. Mijo Mirković“, Croatia  
e-mail: darko.etinger@unipu.hr

Aljosa Vitasović  
Juraj Dobrila University of Pula  
Department of Economics and Tourism „Dr. Mijo Mirković“, Croatia  
e-mail: avitasov@unipu.hr

Jusuf Šehanović  
Juraj Dobrila University of Pula  
Department of Economics and Tourism „Dr. Mijo Mirković“, Croatia  
e-mail: jusuf.sehanovic@unipu.hr

Abstract
E-learning as a network-enabled transfer of skills and knowledge is powered by solutions originating from both commercial and non-commercial producers. As the purchase and maintenance of computer software typically accounts for 20 percent of an institution’s IT budget, higher education institutions can no longer afford to limit themselves to using traditional fee-based proprietary software. When appropriately managed, the use of open-source software can provide an institution with more effective and less costly software solutions. This paper deals with implementation of open source software solutions based on open standards in e-learning environment of higher education institutions. Open standards ensure interoperability that is critical to any distributed e-learning system.

Keywords: open source, e-learning, open standards, education

1. The open source software movement

The impact of open source software (OSS) produced by computer programmers who volunteer in the process of it's production has been studied by legal scholars, economists, sociologists, anthropologists and computer scientists.
The open source software movement is not new. Public benefit corporations such as the California based The Open Source Initiative (OSI) founded in 1998, are actively involved in open source community-building, education, and public advocacy to promote awareness and the importance of non-proprietary software. By definition, open source does not just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Free Redistribution</td>
<td>The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.</td>
</tr>
<tr>
<td>2. Source Code</td>
<td>The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.</td>
</tr>
<tr>
<td>3. Derived Works</td>
<td>The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.</td>
</tr>
<tr>
<td>4. Integrity of The Author's Source Code</td>
<td>The license may restrict source-code from being distributed in modified form only if the license allows the distribution of &quot;patch files&quot; with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.</td>
</tr>
<tr>
<td>5. No Discrimination Against Persons or Groups</td>
<td>The license must not discriminate against any person or group of persons.</td>
</tr>
<tr>
<td>6. No Discrimination Against Fields of Endeavor</td>
<td>The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.</td>
</tr>
<tr>
<td>7. Distribution of License</td>
<td>The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.</td>
</tr>
<tr>
<td>8. License Must Not Be Specific to a Product</td>
<td>The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.</td>
</tr>
<tr>
<td>9. License Must Not Restrict Other Software</td>
<td>The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.</td>
</tr>
<tr>
<td>10. License Must Be Technology-Neutral</td>
<td>No provision of the license may be predicated on any individual technology or style of interface.</td>
</tr>
</tbody>
</table>

Table 1. The distribution terms of OSS (http://www.opensource.org/docs/osd)
Edwards (2005.) points out that the existence and continued development of OSS is intriguing. Programmers situated all over the planet contribute to developing software, which sometimes reach a level of quality where it becomes the dominating software package in its category. Projects such as SourceForge (http://sourceforge.net), the world's largest OSS development web site, with more than 230,000 registered software projects and more than 2 million registered users as of February, 2009 are good examples of the potential of the open source movement. The open source system exploits the potential of all the users. According to Johnson (2001.), this can (but need not) result in only low costs being borne. The benefits of open source code is that more information is being used contributing to it's value in the sense that users can directly contribute to the community. When source code is unavailable publicly, the human capital and insight present in the community as a whole cannot be harnessed. Open source code potentially allows the entire Internet community to use its combined programming knowledge, creativity and expertise.

The success open systems like Linux, coupled with the decisions of major corporations such as IBM and Sun Microsystems to open their source code, led to a production of a wide variety of useful and reliable software.

2. Basic characteristics of open source and proprietary software as foundation for e-learning environment construction

There are several definitions of the term e-learning and it can be observed by many points of view. E-learning is essentially the network-enabled transfer of skills and knowledge. It is powered by solutions originating from both commercial and non-commercial producers.

Trappler (2009.) points that software provides the foundation of an institution’s IT infrastructure, and the purchase and maintenance of computer software typically accounts for 20 percent of an institution’s IT budget. Higher education institutions can no longer afford to limit themselves to using traditional fee-based proprietary software. When appropriately managed, the use of open-source software can provide an institution with more effective and less costly software solutions. To elaborate this statement, the following comparison between OSS and proprietary software is given:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Open-Source Software</th>
<th>Proprietary Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitness of Purpose</td>
<td>• no license fee, evaluation licenses at no cost, no expiration date&lt;br&gt;• participation in the OSS product’s community can enable an organization to influence the direction of the product in a way that aligns with their own needs. &lt;br&gt;• not always obvious how to identify and obtain a particular OSS solution; additional internal resources may be required.</td>
<td>• evaluation licenses have a limited duration and sometimes an associated fee&lt;br&gt;• possible change of direction of a product or discontinuing due to lack of profitability, a merger/acquisition, or bankruptcy&lt;br&gt;• clearer avenues for identifying and obtaining PS products than with OSS</td>
</tr>
</tbody>
</table>
| Quality          | development isn’t well managed; more mature OSS projects tend to be better managed  
|                 | no one is accountable for code quality but often more reliable; continually peer-reviewed through use of the software  
|                 | the need to proactively maintain knowledge of updates and upgrades  
|                 | identifying a reliable source for version updates can be challenging without third-party support  
|                 | development often more effectively managed  
|                 | developers ensuring code is error-free, but rarely a PS product never requires patches or bug fixes  
|                 | vendors proactively provide notification of updates and upgrades  
|                 | usually a sole and trusted source of version updates for its product.  
| Reliability     | OSS products require sustained community participation from skilled programmers to remain viable  
|                 | while the focus of an OSS product may change direction, a user has the right to remain indefinitely on any version  
|                 | the continued availability of a PS product is based upon its commercial viability.  
|                 | PS vendors may decide to not allow a user to remain on a prior version  
| Security        | the code is open, so all users can view it, which can lead to problems being discovered and fixed more quickly  
|                 | the code is open, so anyone (including malicious users) can view it to determine how to exploit potential vulnerabilities  
|                 | the code is open, so a user can review and determine the actual level of security prior to adopting the product  
|                 | PS vendors can have numerous full-time software developers who are paid to ensure that code is secure  
|                 | the code is not open, so it is more difficult for malicious users to determine how to exploit potential vulnerabilities  
|                 | since PS code is not open, it is not possible for a user to review in advance and determine the actual level of security  
| User Friendliness | its origins in IT infrastructure have given OSS the image of having poor user interfaces “built by IT for IT,” but this is changing as OSS product options expand to applications, including Firefox, OpenOffice, and Ubuntu Linux  
|                 | user friendliness can vary by product, but PS products generally have better user interfaces  

Table 2.: OSS evaluation (Trappler 2009.)

According to Foreshew (2005.), the findings from a survey of more than 100 Australian educational institutions and universities, conducted by e-learning systems provider WebCT found that 89 per cent of respondents consider open-source systems alone too unstable, too costly to maintain and lacking sufficient support to be viable. However Hernandez, Pardo and Kloos (2007.) acknowledge the significant quality increase in open-source e-learning platforms allowing for large-scale e-learning courses with significantly reduced costs.
E-learning applications initiatives by higher education institutions are considering OSS solutions where the software is freely available for delivering education online (Coppola & Neelley, 2004). Siemens (2003) suggests that the benefits of the open source model are increased quality, greater stability, superior performance, and improved functionality. Reduced vendor reliance, reusability, reduced costs, auditability (users validating security), reliability, and rapid fixes to bugs/problems are among other benefits open source model can offer.

In the past several years higher education institutions have initiated the creation of enterprise open source applications such as course management systems and electronic portfolios. These e-learning applications initiatives are initial steps higher education is taking to move away from proprietary software toward open source (Koohang and Harman, 2005.). With open source, higher education institutions can easily and freely audit their system. The system becomes open, transparent and flexible with ultimate access/control, ownership, and freedom. An open system encourages increased exchange of ideas that advances innovation.

The e-learning movement toward open source model has been evident in the recent years. There are many open source projects dedicated to e-learning. Below is a list of several selected organizations that are involved in helping professors build their e-learning courses:

- Claroline (http://www.claroline.net)
- .LRN Course Management (http://www.collaboraid.biz/products/dotlrn)
- EduZope (http://www.eduzope.org)
- Moodle (http://moodle.org)
- Pachyderm (http://www.nmc.org/projects/lo/pachyderm.shtml)
- Sakai (http://www.sakaiproject.org)
- Spaghetti Learning (http://www.spaghettilearning.com)
- A Tutor (http://www.atutor.ca)

### 3. Developing virtual learning environment (VLE) using OSS solutions based on Open Standards

E-learning tools are making education better with benefits such as reduced administrative workload, increased collaboration between students and pedagogical flexibility for an increasingly diverse student population.

A polarized debate between those who support open source VLEs and those who support commercially-developed solutions is present. As Stephenson (2006.) indicates, a solution in between open source and proprietary systems is possible as they are not mutually exclusive.

Commercial e-learning solutions are often highly scalable, extensible and comprehensive products, placed in the best position to incorporate quality and performance, software maintenance, a formalized feature enhancement process, customer support, and the aftermarket services required to support an enterprise-scale e-learning solution. The opportunity exists to build on these by providing a broader range of functional alternatives.
that integrate with and complement commercial e-learning platforms. Several industry's leading companies have already embraced this way that merges OSS with PS ia a form of Open Systems.

In Open Systems, portions of the vendor code are exposed to academic open source communities via standards based APIs so that they can modify components and add their own (or third party) extensions to the application’s core teaching and learning capabilities. Maintaining the mission-critical components of the overall e-learning platform requires development and support resources that commercial vendors are best positioned to deliver. Rather than struggling to build and sustain the full e-learning infrastructure and platform, the open systems approach allows open source communities to focus their efforts on developing the specialised teaching and learning extensions that fully leverage their pedagogical experience and innovative ideas. This added-value development approach allows Open Source developers to build on the strengths of commercial systems, combining the benefits of both for the best overall solution (Stephenson 2006.).

Open systems are an attractive alternative to the current bickering that characterises the e-learning marketplace. Commercial and open source vendors can work in tandem to offer the flexibility for customization and extension that institutions find attractive, typically at a lower cost of ownership and without the inherent risks of pure open source projects. Open systems also provide the reliability, completeness, and security of a thoroughly tested and professionally supported commercial solution.

In addition to Open Systems, a special care must be taken of Open Standards. Open Standards ensure interoperability that is critical to any distributed e-learning system. The open source community must implement open standards in creating open source e-learning systems. Several models of open standards are provided:

1. IEEE LO Metadata (LOM) Learning Technology Standards Committee (LTSC) P1484 (http://ltsc.ieee.org)
2. IMS (Instructional Management System) Global Learning Consortium (http://www.imsglobal.org)
3. Advanced Distributed Learning (ADL) Initiative - Shareable Courseware Object Reference Model (SCORM) (http://www.adlnet.org)
4. PROMETEUS: PROmoting Multimedia Access to Education and Training in EUropean Society (http://www.prometeus.org)

Open Standards is more than just a specification. The principles of Open Standards are:

1. Availability - Open Standards are available for all to read and implement;
2. Maximize End-User Choice - Open Standards create a fair, competitive market for implementations of the standard. They do not lock the customer in to a particular vendor or group,
3. No Royalty - Open Standards are free for all to implement, with no royalty or fee. Certification of compliance by the standards organization may involve a fee;
4. No Discrimination - Open Standards and the organizations that administer them do not favor one implementor over another for any reason other than the technical standards compliance of a vendor's implementation. Certification organizations must provide a path for low and zero-cost implementations to be validated, but may also provide enhanced certification services;
5. Extension or Subset - Implementations of Open Standards may be extended, or offered in subset form. However, certification organizations may decline to certify subset implementations, and may place requirements upon extensions;
6. 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.

4. Conclusion

A higher education E-learning environment is a dynamic and constantly evolving field. IT's costs tend to be stochastic without proper strategies of new technologies implementation. For a long-term period, I find the following to be the optimum model of implementation to take in consideration.

As many authors and significant leading companies in the field of e-learning innovations show, OSS solutions based on Open Standards in e-learning environment of higher education institutions combined with existing PS solutions that ensure interoperability and expansion opportunities crucial for a long-term sustainability of VLEs.

Implementation of OSS solutions based on Open Standards in e-learning environment of higher education institutions do offer an effective ground in terms of purpose, quality, reliability, security and user friendliness.

Literature


