Tampereen teknillinen yliopisto. Ohjelmistotekniikan laitos. Raportti 21
Tampere University of Technology. Department of Software Systems. Report 21

Gregorio Robles, Jesús González Barahona,
Wouter Tebbens & Imed Hammouda (eds.)

Proceedings of FLOSSEdu 2012: FLOSS Education - Long-term Sustainability

TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Tampereen teknillinen yliopisto. Ohjelmistotekniikan laitos. Raportti 21
Tampere University of Technology. Department of Software Systems. Report 21

Gregorio Robles, Jesús M. González Barahona,Wouter Tebbens & Imed Hammouda (eds.)

Proceedings of FLOSSEdu 2012: FLOSS Education – Long-term Sustainability

Tampere University of Technology. Department of Software Systems
Tampere 2012

ISBN 978-952-15-2938-2
ISSN 1797-836X

Preface

This is a collection of the papers presented at the FLOSS Education: Long-term sustainability (FLOSSEdu 2012). The workshop was co-located with OSS 2012 and was held in Hammamet, Tunisia on September 13, 2012.

The goal of the workshop was to bring together a community that is interested in education and Free/Libre and Open Source Software (FLOSS). Its importance can be understood as learning in any environment has undergone major changes in recent times, and many of those changes are closely related or are similar to processes or solutions found traditionally in Free/Libre and Open Source (FLOSS) environments: peer to peer relationships, telematic support, use of new technologies, etc. So, the purpose of this workshop was to bring together free software experts to discuss challenges that we face in the educational world at present and that we are likely to face in the future and how they can be undertaken from a FLOSS perspective.

The program contained 4 technical presentations discussing topics related to education, from various angles and perspectives. For the selection of the presentations, authors had to submit contributions that have been peer reviewed by at least two members of the programme committee.

We would like to thank the members of the organizing committee and the program committee for their effort. We also appreciate the contribution of the authors of papers submitted. We are also very thankful with the OSS 2012 organizers, and especially with the local support received during the workshop.

We hope that you enjoy the workshop!

Tampere, October 2012

Gregorio Robles, Jesús M. González Barahona, Wouter Tebbens and Imed Hammouda

Organizing Committee

Jesús M. González-Barahona (Universidad Rey Juan Carlos, Spain)

Wouter Tebbens (Free Knowledge Institute, The Netherlands)

Gregorio Robles (Universidad Rey Juan Carlos)

Programme Committee

Hinde ten Berge (Free Knowledge Institute, The Netherlands)

Jesús M. González-Barahona (Universidad Rey Juan Carlos, Spain)

Israel Herraiz Tabernero (Universidad Politécnica de Madrid, Spain)

Daniel Izquierdo Cortázar (Bitergia, Spain)

David Jacovkis (Free Knowledge Institute, The Netherlands)

Gregorio Robles (Universidad Rey Juan Carlos)

Wouter Tebbens (Free Knowledge Institute, The Netherlands)

Sponsorship

This workshop is sponsored by the Free Technology Academy (1) and the eMadrid Network of Excellence (2).

(1) http://learning.libresoft.es/images/flossedu/fta.png

(2) http://www.emadridnet.org

Contents

Utilization of OSS Virtual Machines for the Hands-on Training Environment

Jun Iio[1]
1 Mitsubishi Research Institute, Inc.
Future Designs and Analysis Center
2-10-3 Nagatacho, Chiyoda-ku, Tokyo 100-8141 Japan
iiojun@mri.co.jp

Abstract. Now the computing capabilities have improved drastically, and this has made it easier for preparing the environment to run virtual machines with less stress. I report a proposal to try using virtual machines, which are commonly used for servers, for an educational use. In our activities on open-source education, we offer several hands-on training experiences. And we have some problems to prepare the computing environment for doing such hands-on training. To solve these problems, I propose that the computing environment should be prepared using the open-source virtual machines. In this paper, the background of preparing OSS virtual machines for the hands-on training environment, its pros-and-cons, and my future plan are discussed.

1 Introduction

Computing power has been growing drastically along with Moore's Law, and now the calculating capabilities have improved dramatically. This growth has made it easier for utilizing the environment to run a number of virtual machines on a physical machine, simultaneously. Especially in a server environment, the use of virtual machine has become widely common. In addition, the utilization of virtual machines is also focused in the academic field, such as the university cloud-computing facilities. Furthermore, teaching the cloud computing paradigm becomes more important in the curricula of schools of computer science and information systems [1]. In this paper, a practical usage of virtual machines in order to prepare hands-on training environment is reported.

2 Problems

I give several lectures on information technology engineering as a part of a government-sponsored program at Tokyo University of Agriculture and Technology (TUAT), since 2009. The lectures of the program include “Special lecture on web-computing” and “Special lecture on open-source software development.” These courses consist not only of classroom lecture but also of hands-on training. Hence, the lectures need to be provided in a computer room, with students sitting in front of each computer connected to the local area network in the university.

Sometimes, students have to work with the computer to try a task given by teachers. For example, in the OSS development course, students can choose some tasks shown in Table 1 for their themes for the hands-on training. To conduct this training in a quick and efficient manner, teachers have to prepare the working environment preliminary.

Table 1. Tasks provided by the textbook [2], as examples of the efficient use of ready-made resources in C-language programming.

Handling command-line option	GUI programming
Data structure	Computer graphics
Database application	Accessing devices
Networking	Encode / decode, text handling
Scientific calculation	Testing framework
Image processing	

The computers implemented in the computer room at TUAT are configured to multi boot[1] environment both with Microsoft Windows and with CentOS Linux. For the OSS development course, CentOS is considered as a good candidate of the hands-on training platform. Therefore, some libraries needed for the hands-on training have to be installed into the CentOS machine image in a master computer for the network booting.

However, there remain some problems. For some libraries, there are differences between the version numbers of the libraries installed and those shown in the textbook. They are due to the difference of the platforms. Practically, we prepared the latest CentOS platform for the training course, but the textbook supposes that the readers should try the practices written in the textbook using the latest Ubuntu. Moreover, due to security reason, it is not permitted to install some libraries (e.g. a library offering functions as network sniffer.)

In addition, even if we want to construct a server in the network of TUAT, it requires a complicated procedure, so there remains a problem that we have to use stand-alone version of a content management system (CMS), when students are trying to configure the CMS as part of the hands-on training.

3 Solution using Virtual Machines

To solve the problems previously mentioned, introduction of the work environment using virtual machines is proposed. This solution can not only eliminate a mismatch between practical working environment and ideal one but also provide some

[1] These client terminals are booted via the local area network in the university.

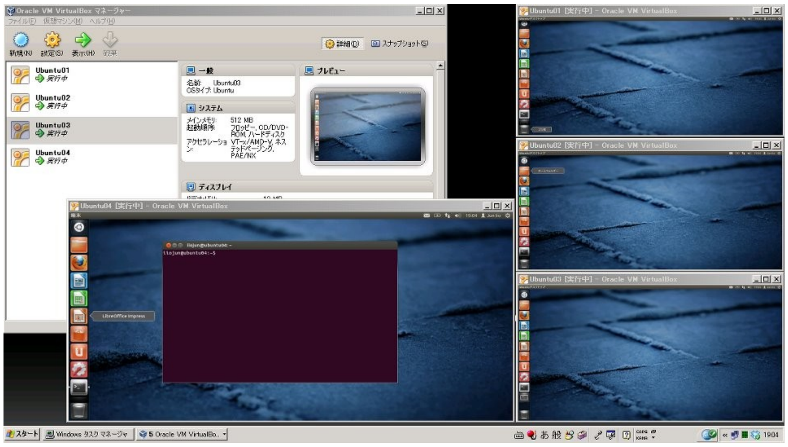
advantages such that teachers can configure the environment remotely and with trials and errors.

3.1 Preparing a Virtual Machine

Preparing a virtual machine follows the procedure below.

1. Start VirtualBox and create a new virtual machine image.
2. Install the latest version (or the version explained in the textbook) of Ubuntu Linux.
3. Keep the OS up-to-date by applying security updates.
4. Install libraries required for the hands-on training.
5. Export the virtual machine image after the configurations are complete.

At the preparation of hands-on training, exported machine images are delivered to the computers used by students. Following these steps offers an optimally-configured environment in safe condition for students to tackle with their training tasks. If the newer updates are provided after the time when machine image was created, the update will be applied as necessary.



名前: Ubuntu03
OSタイプ: Ubuntu

メインメモリ: 512 MB
起動順序: フロッピー, CD/DVD-ROM, ハードディスク
アクセラレーション: VT-x/AMD-V, ネステッドページング, PAE/NX

```
liojun@ubuntu04:~$
```

Fig. 1. Four virtual machines on VirtualBox running in a Microsoft Windows desktop environment. Each virtual machine is equipped the latest version of Ubuntu Linux and configured suitable for the hands-on training.

This approach has many advantages. Creating images can be done anywhere and anytime, remotely, disconnected from the network of the university. It is independent of the platform implemented in the existing computers. There is a lot of flexibility to configure the environment for the hands-on training. Once the configuration is complete, all you need to do is copy the image into computers for students. The machine image can be saved and can be reused many times.

In addition, running multiple images simultaneously can provide a pseudo-distributed computing environment in a very simple way. Figure 1 shows a snapshot on the Microsoft Windows desktop where the four virtual machines of Ubuntu Linux are running. These virtual machines are connected with a virtual network within a host OS. By using these virtual machines and network, distributed computing framework like Hadoop can be easily tested on one physical computer in order to learn distributed computing technologies.

3.2 Disadvantage of This Solution

There are a few disadvantages of this solution. The major one is the cost to deliver the large size machine image. In general, the size of virtual machine is large enough so that it takes long time to copy the image especially via thin network bandwidth. However, this problem will be solved if the network capability is improved.

The other problem is the fact that any type of libraries can be installed into the virtual machine. This raises a possibility of violation against the security policy for the on-campus network. We should negotiate well with the network administrators to avoid this type of difficulties. The conditions with accessing devices are similar to this issue. If you want to make programs which access hardware devices, negotiation with system administrators would be required.

4 Related work

In order to learn network configuration skills and network security technologies, virtual machines constructing a virtual network environment have been commonly used [3-8], because it requires less cost than preparing a number of physical machines. The cost effectiveness and security issues are the main reasons for the virtual machines being preferred.

Training by using virtual environment is efficient not only in the network technologies but also in software engineering [9]. A potential for virtual machines to be adopted in teaching application development was similarly shown in this paper.

5 Conclusions and Future work

The method of utilizing virtual machines to solve some problems in the OSS technological education has been discussed. Using virtual machines can make it easier to prepare hands-on training environment. Moreover, the virtual network enables for students to experience (pseudo-)distributed computing. Saving a machine image and reusing it easily are also effective advantage of using virtual machines.

The "special lecture on open-source development" in this fiscal year will be carried out from November, 2012, provided as 5 weeks intensive course. In this course, the virtual environment described in this paper will be given to the students, and I will ask them for their comments. As a future work, the virtual machine should be optimized, according to the feedback comment from the students.

References

[1] James P. Lawler, “Cloud Computing in the Curricula of Schools of Computer Science and Information Systems,” *Information Systems Education Journal*, 9(2), pp. 34-41, 2011.

[2] Jun Iio, “Super Linux Programming,” SoftBank Creative, 2011. (In Japanese).

[3] G. Kent Webb, “Using an On-line Virtual Server to Simulate Network Administration,” *Issues in Information Systems*, 4(1), pp. 135-141, 2005.

[4] Ricardo Nabhen and Carlos Maziero, “Some Experiences on Using Virtual Machines for Teaching Computer Networks,” in *Proceedings of Education for the 21st Century — Impact of ICT and Digital Resources*, pp. 93- 104, 2006.

[5] Marco Anisetti, Valerio Bellandi, Alberto Colombo, Marco Cremonini, Ernesto Damiani, Fulvio Frati, Joêl T. Hounsou, and Davide Rebeccani, “Learning Computer Networking on Open Paravirtual Laboratories,” IEEE Trans. on Education, 50(4), pp. 302-311, 2007.

[6] Alok Tripathi, Abhinav Mishra, and G.V. Raghunathan, “A Test Bed for Information Security Skill Development with Virtual Training Environment,” *Management*, 2008.

[7] Fuh-Gwo Chen, Kuo-Yi Chen, and Chin-Yang Lin , “A Cooperative Learning of Computer Networking with Portable Laboratories Using Virtual Machine,” in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology* (ICCSIT), pp. 118-122, 2010.

[8] Christian Willems and Christoph Meinel, “Practical Network Security Teaching in an Online Virtual Laboratory,” in *Proceedings of the 2011 International Conference on Security & Management* (SAM2011), 2011.

[9] Jim Wrubel, David White, and Julia Allen, “High-Fidelity e-Learning: SEI‘s Virtual Training Environment (VTE),” Technical Report, CMU/SEI-2009-TR-005 / ESC-TR-2009-005, 2009.

Free and Open Source Software in Project Based Service Learning

Henry Kim[1], Dan Bolivar[1], Matthew Daye[1], Charles Jhin[1], Jonathan Koch[1],
Matthew Maguire[1], Andrew Sindoni[1], Cedric Strickland[1], Rebekah Deline[2],
Gena Robinson[2], Greg Madey[1]
[1]University of Notre Dame, College of Engineering
Notre Dame, IN 46556 United States
[2]American Red Cross - St. Joseph County Chapter
South Bend, IN 46615 United States
{hkim16, dbolivar, mdaye, cjhin, jkoch2, mmaguir3, asindoni, cstrickl,
gmadey}@nd.edu, { rebekah.deline, gena.robinson}@stjoe-redcross.org

Abstract. This paper introduces software solutions delivered by a Notre Dame Engineering Projects in Community Service (EPICS) team using free and open source software. We first give a brief introduction and background on EPICS at Notre Dame, and the St. Joseph Chapter of the Red Cross. Next, two case studies explore how this enhances the undergraduate students' education, benefits the local community, and remains true to the University's commitment to service. The first study demonstrates the continuation of a previous project and the second study demonstrates the beginnings of a new project. Both studies demonstrate the use of free and open source software in higher education to solve real-world challenges. Finally, the paper shows the current and future goals of the team.

1 Introduction

The Engineering Projects in Community Service (EPICS) program at Notre Dame started in 1997 and was modeled after a similar program of the same name at Purdue University [1]. EPICS at Notre Dame has since evolved to fit the unique needs of its host University to provide valuable service to the local community and allows Notre Dame students to join the popular movement to experiential learning [2]. In fact, one of the core missions of Notre Dame is to "is to create a sense of human solidarity and concern for the common good that will bear fruit as learning becomes service to justice" [3]. This is incorporated into the curriculum through "service learning courses offered by volunteer faculty and graduate student instructors as a technical elective for undergraduate engineering students" [4].

Students take these courses for one credit per semester, usually for about three semesters. The course focuses on two underserved aspects of the computer science curricula, open-source development model and service learning as a way to gain experience by developing solutions to real-world problems [7]. Due to the nature of these problems, students are assigned to a project that can exceed the tenure of any one student. This results in increased emphasis on user and developer documentation, and updating the code repository to ensure consistency and continuity.

2. Case Studies

The American Red Cross is chartered by Congress to "carry on a system of national and international relief in time of peace and apply the same in mitigating the sufferings caused by pestilence, famine, fire, floods, and other great national calamities, and to devise and carry on measures for preventing the same" [5]. As the smallest operational unit in the organization, each chapter works directly with the communities in need.

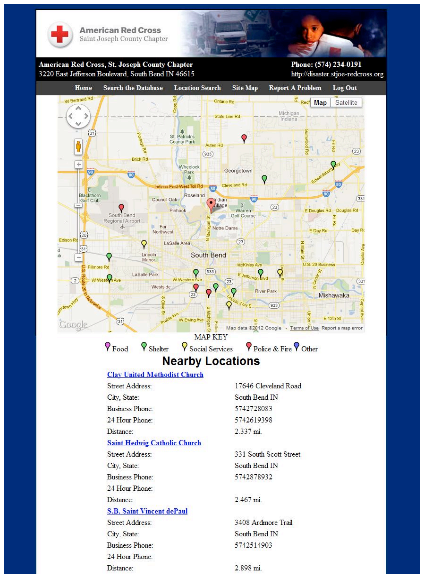
The St. Joseph Chapter of the American Red Cross is responsible for responding to disasters in a seven county area in northern Indiana and southern Michigan. The chapter strives to meet the emergency needs of local disaster victims with vouchers for free food, clothing, temporary shelter, and medications. When in the field, responders need contact information for disaster response resources. These often include organizations that have agreed to provide shelter, food, water, clothing and other basic essentials at no-cost or at a previously agreed-upon rate. Additionally, the Red Cross assesses damage caused by these disasters for planning and decision-making regarding the allocation of relief services.

2.1 Red Cross Disaster Database

The first case study examines the consulting relationship with the Red Cross on developing a disaster resources database. The EPICS team started with a pre-existing Microsoft Access database that was occasionally copied to laptops for use in the field. This set-up caused issues such as duplication of data, discrepancies across the spreadsheets, and a non-optimized table making the data susceptible to loss, damage, and corruption.

In response to these issues, in the fall of 2007, the team originally designed an Oracle database with a web interface solution. Within the semester, the students delivered a prototype that was approved by the chapter leadership. Then, in the spring and fall of 2008, the EPICS team moved the database and web files from Notre Dame servers to the Red Cross hosted by OneEach, Inc. In the process, the team decided to change the database and the web interface to use MySQL and PHP. The source code was produced using various code editors and the documentation was circulated via e-mail.

After deployment, the project was put on hiatus for about one year. But, in the spring of 2012, the project code was brought out of mothballs when the Red Cross found the system needed maintenance. The team evaluated the system, found that the system was repairable, and decided to use most of the previous MySQL and PHP code. The team debugged existing features so that users can add organizations to the database and change user information. Once operational, the team contributed new enhancements requested by the Red Cross such as a location-based search. This was implemented using Google Maps API, which was provided free of charge, to display resources within the database.



American Red Cross, St. Joseph County Chapter
3220 East Jefferson Boulevard, South Bend IN 46615

Phone: (574) 234-0191
http://disaster.stjoe-redcross.org

Nearby Locations

Clay United Methodist Church

Street Address:	17646 Cleveland Road
City, State:	South Bend IN
Business Phone:	5742728083
24 Hour Phone:	5742619398
Distance:	2.337 mi.

Saint Hedwig Catholic Church

Street Address:	331 South Scott Street
City, State:	South Bend IN
Business Phone:	5742878932
24 Hour Phone:	
Distance:	2.467 mi.

S.B. Saint Vincent dePaul

Street Address:	3408 Ardmore Trail
City, State:	South Bend IN
Business Phone:	5742514903
24 Hour Phone:	
Distance:	2.898 mi.

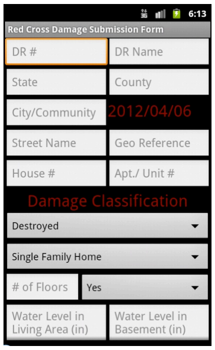
Fig. 1. The Google Maps API used to display selected resources in the disaster database

Due to the nature of this program, team members are re-assigned to various projects as needed and participate for a limited number of semesters. Hence, the team found it convenient to host the projects on Google Code, which provides a code repository, issue tracker, collaborative wiki and other tools. The repository uses Subversion for version control and allows students to collaborate in teams that range in size from three to seven students. The issue tracker allows anybody with a Google account, especially clients, to report an issue, enhancement, or task to improve the website. This allows any student to read, complete, and update that issue. The wiki allows students to share development methods, meeting notes, and team goals. This helps future students determine the reasoning for coding decisions and how a particular piece of code works. In exchange for these tools, the source code and user documentation for this project was made open to the public under GNU GPL v3.

2.2 Red Cross Damage Assessment

Another mission for the St. Joseph County Chapter of the Red Cross is to conduct disaster assessment operations. Currently, representatives record data on paper forms. The second case study began when the chapter requested that the EPICS team develop a solution that automates this process.

The EPICS team began by developing a mobile application on both Android OS and Apple iOS and a corresponding website to utilize the data. Various parts of these mobile apps and website were developed as part of either summer research or Mobile Application Development course. Both of these development method emphasized independent learning into the curriculum. Hence, the students rely more on information found on online manuals, references and forums during development. The Android version is coded in Java using the Eclipse IDE with the Android Development Add-on. The Apple app is coded in Objective C using the Xcode editor and its development tools.



Red Cross Damage Submission Form

DR #	DR Name
State	County
City/Community	2012/04/06
Street Name	Geo Reference
House #	Apt./ Unit #

Damage Classification

Destroyed ▾

Single Family Home ▾

# of Floors	Yes ▾
Water Level in Living Area (in)	Water Level in Basement (in)

Fig. 2. Screenshot of the Android application.

A corresponding website queries the database and allows for the data to be sorted and managed electronically. The functionality of the site is implemented in PHP and utilizes a MySQL database. The mapping utility is implemented using the Google Maps API. The website should eventually include functionalities similar to that described by Zhai et al. to allow for crowd-sourcing and citizen reports.[6]

3. Summary & Conclusions

The PHP and MySQL disaster resources database, which has been slowly evolving for the past five years, has met the expressed needs of the client. The team looks forward to continued collaboration with the Red Cross chapter on their changing needs.

One idea that has been suggested by the chapter is the expansion of the program to other chapters, starting with other regional chapters, and potentially reaching the national level. Future teams may have the opportunity to help pitch this idea to the regional leadership in Indianapolis, and from there, to other chapters in the nation.

The utilization of mobile devices in disaster assessment will require more key features such as storing persistent data and auto-fill functionality. It remains a project for future EPICS students to develop, test, and deliver this solution to the St. Joseph County chapter of the Red Cross.

The unifying themes present in these student projects include: 1) Open Source Software in use for humanitarian purposes and 2) Open Source Software in use for service-based learning projects by engineering students.

4. Acknowledgement

We would like to thank the St. Joseph Chapter of the American Red Cross for their cooperation in developing this system. In particular, we would like to thank John Pinter of the St. Joseph Red Cross for his support and feedback throughout the software development process of this project.

This project was supported in part by a "Rodney F. Ganey, Ph.D. Collaborative Community-Based Research Grant" from the University of Notre Dame Center for Social Concerns. Additional support was provided by the Center for Research Computing and the National Science Foundation under grant #1063084.

5. References

[1] Madey, G., Freeland, C., and Brenner, P., "A Service Learning Program for CSE Students", 35th ASEE/IEEE Frontiers in Education Conference Indianapolis, IN, 2005

[2] Carberry, A., “Work in Progress – Assessing the Engineering Service Students’ Characteristics”, 40th ASEE/IEEE Frontiers in Education Conference, Washington, DC, 2010

[3] “Mission Statement,” University of Notre Dame, Notre Dame, IN http://www.nd.edu/aboutnd/mission-statement/, 2010

[4] Brenner, P., Schroeder, M., and Madey, G., “Student Engineers Reaching Out: Case Studies in Service Learning and a Survey of Technical Need”, 37th ASEE/IEEE Frontiers in Education Conference Milwaukee, WI, 2007

[5] “Congressional Charter of the American National Red Cross,” American Red Cross, Washington, DC http://www.redcross.org/www-files/Documents/Governance/charter.pdf, 2010

[6] Zhi Zhai, David Hachen, Tracy Kijewski-Correa, Feng Shen, Greg Madey, Citizen Engineering: Methods for “Crowd Sourcing” Highly Trustworthy Results. Submitted to HICSS-45, under review.

[7] Morelli, R., De Lanerolle, T., “The Humanitarian FOSS Project: Building Free Open Source Software for Society”, www.hfoss.org, 2008.

Learning through analysis of coding practices in FLOSS projects

Jonas Gamalielsson, Alexander Grahn and Björn Lundell

University of Skövde, Skövde, Sweden,
{jonas.gamalielsson,alexander.grahn,bjorn.lundell}@his.se

Abstract. In recent years there has been increased interest in education related issues in FLOSS research. Through interaction with FLOSS projects that are deployed in a variety of usage contexts, students get unique opportunities for learning about development practices. In line with this, we propose an approach for analysing the relationship between coding policy and coding practice in FLOSS projects, intended to be part an assignment in a FLOSS development course. More specifically, the approach focuses on adherence to coding standards with respect to code commenting. The approach is demonstrated and applied on the PHP-based CMS tools Wordpress, Joomla, and Drupal, which are all provided as FLOSS projects.

1 Introduction

This paper draws from experience of offering FLOSS (Free/Libre Open Source Software, hereafter referred to as OSS (Open Source Software)) courses, and relates to project courses in OSS development aimed to be offered both at undergraduate as well as advanced level, and offered in a variety of different contexts including campus course, distance course, and contract teaching in professional organisations. In particular, it addresses an assignment with a focus on coding policy and practice in larger OSS projects.

In recent years there has been increased interest in education related issues in OSS research as evidenced by a wide variety of studies on various topics such as undergraduate research opportunities in OSS (Boldyreff et al., 2009), learning through practical involvement in OSS (Berdou, 2007; Kilamo et al. 2010; Lundell et al., 2007), teaching experiences in OSS courses (German, 2005), and learning through mining of OSS project metadata (Squire and Duvall, 2009). It has been claimed that literature aimed at teaching software engineering theory often use toy examples, and that “we need to find innovative ways of integrating project work in curricula” (Ghezzi and Mandrioli, 2005). Therefore, in courses on OSS development it would be important for students to increase their understanding of and skills regarding project work in large OSS projects. One aspect to consider in such projects is coding practices. By studying coding practices in large OSS projects, students can: 1) learn how to characterise coding practices, 2) learn about actual coding practices and how practice relates to coding policies, and 3) learn how to best contribute to OSS projects in terms of coding practices.

As part of an assignment in OSS development courses we propose an approach for analysis of coding practices. In particular, the approach addresses practices in code commenting. Comments are part of the documentation of OSS projects, and are interesting to study since it has been noted that improved documentation can contribute to increased participation in an OSS project (Mockus et al., 2002). Further, the lack of documentation and updated documentation is a problem in many OSS projects, and one reason for this is that developers are often not required to provide projects with documentation (Levesque, 2004). Although many projects have policies in the form of coding standards, which contributors are expected to adhere to, there is limited research on the actual adherence to coding standards in large OSS projects.

Earlier studies have explored the growth of documentation and code over time in OSS projects (Fluri et al. 2007; Jiang and Hassan, 2006; Schreck et al., 2007). Further, the comment density of code in OSS projects has been explored (Arafat and Riehle, 2009; Elish and Offut, 2002). We note that there is currently a lack of research on how code commenting practices relate to coding standards. One exception to this is a limited study on how Java classes adhere to certain standard coding practices (Elish and Offut, 2002). Our proposed approach is more comprehensive in that it is based on the actual guidelines of the projects and that it provides detailed information about the occurrence of different kinds of coding errors over time.

2 Research approach

To detect the number of violations of a defined coding standard with respect to code commenting, PHP_CodeSniffer[1] was used in combination with a custom script, which collects the last revision each month from the SCM (Software Configuration Management system) of an OSS project. A custom standard was created to be used by PHP_CodeSniffer, and this was based on the current coding standards for a project.

To demonstrate the approach we decided to apply it on the three PHP-based CMS tools WordPress[2], Joomla[3], and Drupal[4], which are all provided as OSS projects. These tools were chosen since they are the three most used open CMS tools (Shreves, 2010), and that they are deployed in a variety of usage contexts for important systems in private and public sector organisations. The project data in the SCM repositories and current coding standards were collected on 1 June 2012 for the three tools from the locations stated in table 1. All files with a ".php" extension were analysed for each of the three tools. Further, in addition to quantitative processing of project data we recognise that students using the approach in a course context will scrutinise a variety of additional data sources (e.g. forums, mailing lists, documentation, blogs, and other sources related to the project being analysed), which promotes a more in-depth learning experience.

[1] http://pear.php.net/package/PHP_CodeSniffer
[2] http://wordpress.org/
[3] http://www.joomla.org/
[4] http://drupal.org/

Table 1. Location of SCM repositories and coding standards

Tool	SCM	Coding standard
Wordpress	http://core.svn.wordpress.org/trunk/	http://codex.wordpress.org/WordPress_Coding_Standards
Joomla	http://joomlacode.org/svn/joomla/development/trunk/	http://docs.joomla.org/Coding_style_and_standards
Drupal	git://git.drupal.org/project/drupal	http://drupal.org/coding-standards

3 Results

To demonstrate the proposed approach for analysis of coding practices, we here show results from the application to three OSS projects. Figure 1 provides an overview of the degree of adherence to the coding standards for the Wordpress, Joomla and Drupal projects by showing the average number of commenting errors per file for different revisions from the start of each project until end of May 2012. It can be observed that there is an increasing error rate for Wordpress (green trace in Figure 1) until the beginning of 2010, when error rate begins to drop. Further, it can be noted that the long-term trend in Joomla (red trace) is a decreasing error rate. Drupal (black trace) exhibits a more fluctuating error rate with a notable peak in early 2011. There may be various reasons for the variations in error rate such as external events affecting a project and changes in working practice within a project. As an example, we conjecture that the peaks in mid 2008 and early 2011 for Drupal may be related to the start of work on Drupal versions 7 and 8, respectively.

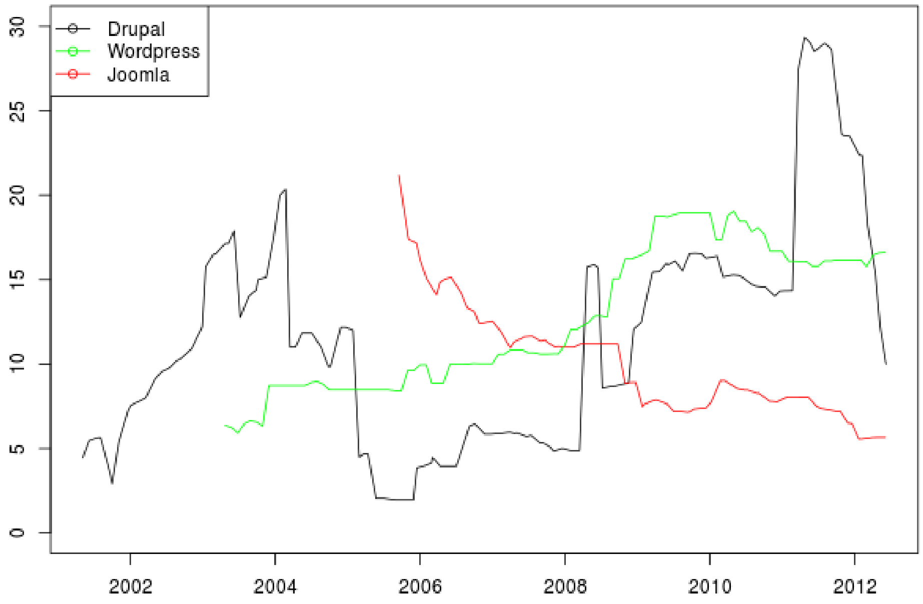


Fig 1. Average number of commenting errors per file over time

Table 2 shows the 12 most frequently occurring types of commenting errors in the Wordpress project over all revisions since the start of the project. The most common error is missing function comment, followed by missing tag in file comment, which together comprise 38.6% of all errors. For Joomla and Drupal the corresponding two most frequently occurring types of errors are related to incorrectly written function comments (representing 34,8% for Joomla and 38,0% for Drupal), rather than missing function comments or missing file comment tags as for Wordpress.

Table 2. Most frequently occurring types of commenting errors in Wordpress

%	Type	Description
21,4	FunctionComment.Missing	The function definition is not documented
17,2	FileComment.MissingTag	The file doc block comment is missing a required tag
13.3	FunctionComment.WrongStyle	The function comment is written with the wrong comment-style (e.g. // instead of /**)
11,6	FunctionComment.MissingReturn	No @return tag in the function comment
7,4	Class.MissingTag	The class definition is not documented
6,2	Function.MissingParamComment	There is a empty comment for the parameter
5,0	Function.MissingParamTag	There is no comment for the parameter
3,7	Function.MissingVersion	Missing PHP version in the file comment
2,0	FileComment.Missing	There is no comment that documents the file
2,0	FunctionComment.ParamNameNoMatch	The name used in the param does not match the actual name of the parameter in the code
1,0	ClassComment.Missing	There is no comment that documents the class
0,8	FileComment.WrongStyle	Invalid type of file comment (e.g. // instead of /**)
8,4	*15 other types of errors*	

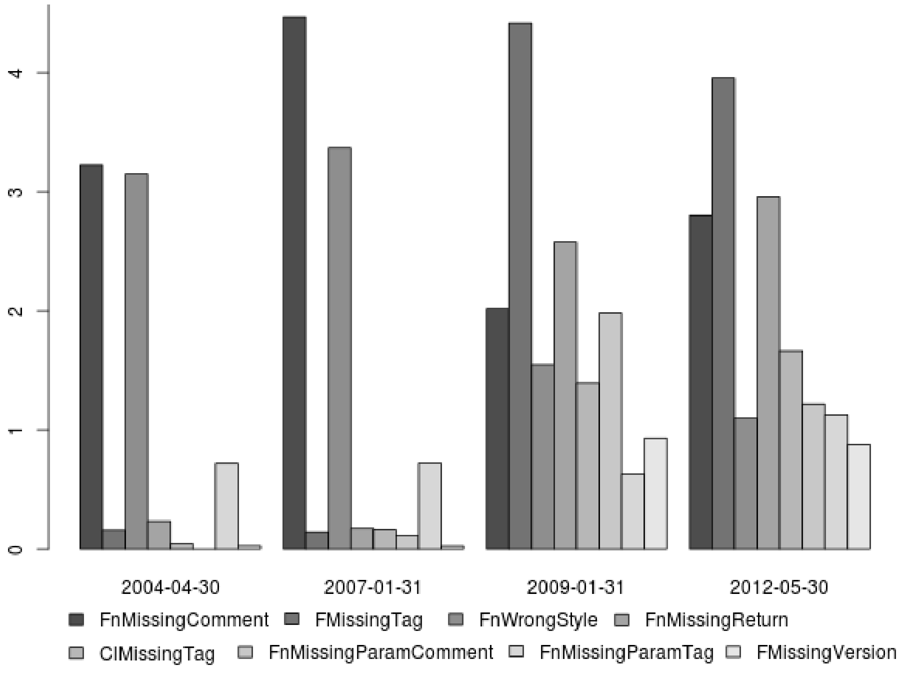


Figure 2. Error rate for different error types over different revisions of Wordpress

Figure 2 shows the error rate for the eight most common types of errors in table 2 for four specific revisions of Wordpress as a characterisation of changes in working practice with respect to code commenting. Missing function comments and wrong commenting style were the most common errors during the first two revisions in Figure 2, whereas missing tag in file comment and missing return tag in function comments are the dominating error types for the two later revisions. We note that a larger number of error types significantly contribute for the two later revisions compared to the two first revisions in Figure 2.

4 Conclusion & discussion

In this paper we have proposed an approach for analysis of coding practices in OSS projects as part of an assignment in an OSS development project course. Through use of such an approach students will be exposed to coding practices and can gain valuable insights from large and widely deployed OSS projects. The approach was demonstrated by applying it to the Wordpress, Joomla and Drupal projects. The focus was on practices regarding commenting, and a characterisation of the adherence to the coding standards of the three projects was presented. Although the focus was on commenting, the approach can easily be extended to cover all relevant aspects of coding standards, which would further promote learning about coding practices in the context of OSS development project courses.

There are different views on the need for code commenting in communities of different OSS projects. For example, some contributors in the Wordpress community advocate use of commenting practices whereas others find it unnecessary. It has for example been claimed that the "inline documentation effort is headed for failure unless all of the core developers understand that inline documentation is not only important, but required. Without it, you have a situation, where some of the code has inline documentation and most doesn't." (Santos, 2008), whereas others claim that comments "are a good way to help a new developer learn the internals" (Merrill, 2006). On the other hand, it has also been claimed that it is "good to have standards. It's not good to adhere to them too rigidly" (Wood, 2009). Further, Santos (2008) note that you "can't force anyone to do anything in an open source community. Enough people do great things that I doubt inline documentation is a major boon or thorn to anyone. Depressing, but I'll rather be coding myself thank you". However, it should be noted that adherence to coding standards may be dependent on an individual's motivation for participation. In addition to volunteer based code contributions, a substantial amount of contributions origin from professionals employed in commercial companies, which motivates analysis of different types of OSS projects.

The approach can be used in different types of course contexts, including campus courses, distance courses, and contract teaching scenarios. Especially for the two latter, the proposed approach may be particularly interesting from a life-long learning perspective. Results from previous research on professionals and their involvement in Open Source show that almost all participating in OSS development projects "cited skills development as an important outcome of participating" (Lundell et al., 2010), and that such skill development "happened through both detailed scrutiny of other people’s contributions and the rigours of writing and exposing their own contributions to scrutiny” (Lundell et al., 2010). With the availability of mature and widely deployed OSS projects, organisations and individuals involved in education obtain new opportunities on how to gain insights into development practices used in a variety of large and mission-critical systems. This in turn imposes new challenges for any organisation involved in offering courses in order to adapt to evolving needs for life-long learning as a long-term strategy for promoting increased innovation. With demands for increased flexibility in how courses are organised and conducted, we suggest that the proposed approach has an important role to play.

References

Arafat, O., Riehle, D.: The Commenting Practice of Open Source. Companion to the Proc. of the 22nd Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA Onward! 2009), ACM press, pp. 857-864 (2009)

Berdou, E.: Learning and the imperative of production in Free/Open Source development. In Feller et al. (Eds.), Open Source Development, Adoption and Innovation, Springer, Berlin, pp. 235-240 (2007)

Boldyreff, C., Capiluppi, A., Knowles, T., Munro, J.: Undergraduate Research Opportunities in OSS. In Boldyreff et al. (Eds.): OSS 2009, IFIP AICT 299, pp. 340-350 (2009)

Elish, M., Offutt, J.: The Adherence of Open Source Java Programmers to Standard Coding Practices. In Proc. of the 6th IASTED International Conference Software Engineering and Applications, pp. 193-198 (2002)

Fluri, B., Wursch, M., Harall, G.: Do Code and Comments Co-Evolve? On the Relation Between Source Code and Comment Changes. In Proc. of the 14th Working Conference on Reverse Engineering Software Quality Journal (WCRE 2007), pp. 70-79 (2007)

German, D.: Experiences teaching a graduate course in Open Source Software Engineering, In Scotto, M. and Succi, G. (Eds.) Proceedings of the First International Conference on Open Source Systems, Genova, Italy, 11-15 July 2005, pp. 326-328 (2005)

Ghezzi, C., Mandrioli, D.: The Challenges of Software Engineering Education. In Proc. of the 27th international conference on Software Engineering (ICSE 2005), 15-21 May 2005, St. Louis, Missouri, USA, pp. 637-638 (2005)

Jiang, Z. M., Hassan, A. E.: Examining the evolution of code comments in PostgreSQL. In Proc. of the 2006 International Workshop on Mining Software Repositories (MSR 2006), pp. 179–180 (2006)

Kilamo, T.: The Community Game: Learning Open Source Development Through Participatory Exercise. In Proc. of AMT 2010. Tampere, Finland, October 2010, ACM Press (2010)

Levesque, M.: Fundamental issues with open source software development. First Monday 9(4) (2004)

Lundell, B., Lings, B., Lindqvist, E.: Open source in Swedish companies: where are we? Information Systems Journal 20(6), 519-535 (2010)

Lundell, B., Persson, A., Lings, B.: Learning Through Practical Involvement in the OSS Ecosystem: Experiences from a Masters Assignment. In Feller et al. (Eds.), Open Source Development, Adoption and Innovation, Springer, Berlin, pp. 289-294 (2007)

Merrill, S.: [wp-hackers] Inline Documentation, http://lists.automattic.com/pipermail/wp-hackers/2006-February/004936.html, 15 February 2006, accessed 13 June 2012 (2006)

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

Santos, J.: [wp-hackers] Inline Documentation Effort was a Failure, http://lists.automattic.com/pipermail/wp-hackers/2008-May/020214.html, 22 May 2008, accessed 13 June 2012 (2008)

Schreck, D., Dallmeier, V., Zimmermann, T.: How documentation evolves over time. In Proc. of the Ninth International Workshop on Principles of Software Evolution (IWPSE 2007), pp. 4-10 (2007)

Shreves, R.: Open Source Market share report, http://www.waterandstone.com/downloads/2011OSCMSMarketShareReport.pdf, accessed 13 June 2012 (2011)

Squire, M., Duvall, S.: Using FLOSS Project Metadata in the Undergraduate Classroom. In Boldyreff et al. (Eds.): OSS 2009, IFIP AICT 299, pp. 330-339 (2009)

Wood, S.: Wordpress Coding Standards, http://www.wptavern.com/forum/general-wordpress/1121-wordpress-coding-standards.html, 17 December 2009, accessed 13 June 2012 (2009)

Integrating Heterogeneous Learning Spaces through Portable Education Portfolios

Veerendra Kumar D.[1], Imed Hammouda[1] and Andreas Meiszner[2]
[1] Department of Software Systems
Tampere University of Technology
Tampere, Finland
firstname.lastname@tut.fi
[2] United Nations University | UNU-MERIT | CCG
The Netherlands
andreasmeiszner@gmail.com

Abstract. The digital age has given rise to new kinds of learning spaces, where learners engage in different kinds of learning activities. A key challenge is how to systematically collect those heterogeneous learning activities into well organized education portfolios. In this paper, we propose the use of Portable Education Portfolios (PEPs) to record and organize learners' experiences. A prototype environment for PEPs has been developed to integrate together open source project contributions and formal course wok into unified education portfolios.

1 Introduction

The use of education portfolios has started in late 1980s to provide tangible evidence of learner achievements [Mil05]. The goal of education portfolios is to represent and certify different artifacts or learning activities achieved by learners. Portfolio techniques are generally used by disciplines such as history and science to promote critical thinking which assesses learner's progress. One popular adoption of the portfolio concept in universities has been the use of electronic portfolios (i.e. e-Portfolios) which allow students to track and record their progress throughout their formal studies [Bha07].

With the success of Open Source Software (OSS) development model, however, there has been increased interest in open education beyond the doors of formal educational institutions. Open education adapts many of the open source principles such co-development and reuse of artifacts. Accordingly, a number of open educational frameworks (e.g. openSE - open education framework for computer science and software engineering [openSE]) have emerged to provide learners with different kinds of open learning projects, allowing learners to achieve recognitions based on given reputation systems.

Furthermore, from open source perspective, there have been calls to integrate open source project contributions, as a new kind of learning activity, into learning portfolios. In this regard, open source projects are seen as a new kind of learning space. Despite the recognized need, no good solutions for such integration exist yet.

In this position paper, we introduce the concept of portable education portfolios (PEPs) as a mechanism to collect different kinds of learning activities, including open source project contributions, into one scalable and secure space. We also report on the early experimental application of PEPs within the openSE environment. The approach has been developed in the context of a course on open source software development, given at Tampere University of Technology.

2 Background

A learning space [Obl06] is an environment where learners engage in learning activities, probably interacting with other learners. Traditionally, learning spaces play an important role in supporting and encouraging students by providing adequate mechanisms and tools. Learning spaces can either be physical and formal, such as classrooms and laboratories or virtual and informal such as online and networked courses [Chr05]. With the advent of ICT, new kinds of learning spaces are emerging, such as online certification programmes and community-driven development projects.

Furthermore, learning has become an activity that is no longer set within programmed schedules and slots. In modern societies, citizens may choose to engage in a life-long learning process (LLLP) [Asp07], consisting of formal, non-formal, and informal learning activities.
Formal learning activities: Learning that belong to formal education and can be achieved through universities, colleges and schools.
Informal learning activities: Learning that occurs in a variety of places, such as at home, at work, through daily interactions, and by means of shared relationships among members of society.
Non-formal learning activities: Learning that involves workshops, community courses, interest-based courses, short courses, or conference style seminars.

From a learner perspective, all the above activities complement each other and contribute to the overall learning experience, despite the fact that those activities are carried out within different kinds of learning spaces. Documenting and organizing those heterogeneous activities into proper education portfolios may be a good vehicle for recording recognition and showcase expertise. This is important for job search for example. There are different types of education portfolios tailored to different purposes [Hen95]. The main ones include documentation portfolios, process portfolios, showcase portfolios, evaluation portfolio, and composite portfolio (see Table 1).

Electronic portfolios, known as e-portfolios [Bha07], have been one of the main digital tools to show evidence of learners' achievements in well organized learning portfolios. E-portfolios, however, have been mostly applied to formal education settings. With the rise of open education resources such as open source communities and other online informal programmes, we argue that another generation of education portfolios, called Portable Education Portfolios (PEPs), is needed. The main difference to such earlier e-Portfolio works is perhaps that PEPs are well integrated within a well-defined approach towards open education. Two central questions emerge within PEPs: how to support as many learning spaces as possible in a smooth, highly interoperable way and how to ensure the credibility and soundness of recorded information.

Table 1. Types of education portfolios

Category	**Description**
Documentation portfolio	It represents a gathering of how much work a student has put over time and her improvement during that time. It can comprise almost everything from brainstorming activities to drafts and completed goods. This approach is the best way to track the best and weakest parts of a student work.
Process portfolio	It documents all phases of the learning process and provides a progressive record of the evolution of students. They are particularly useful in documenting the students' overall learning process.
Showcase portfolio	It records the students' very best work. In this portfolio, only accomplished work should be involved.
Evaluation portfolio	It shows the scores of included work. This is not suitable for all works.
Composite portfolio	It has things from all other kinds of portfolios.

3 Portable Education Portfolios

PEPs are portable education portfolios which are used to import learning activities from different learning spaces (see Figure 1). They provide an authenticated way to import the learning activities of learners from separate learning spaces. It is a system to provide authenticated or certified details of learners to their portfolio.

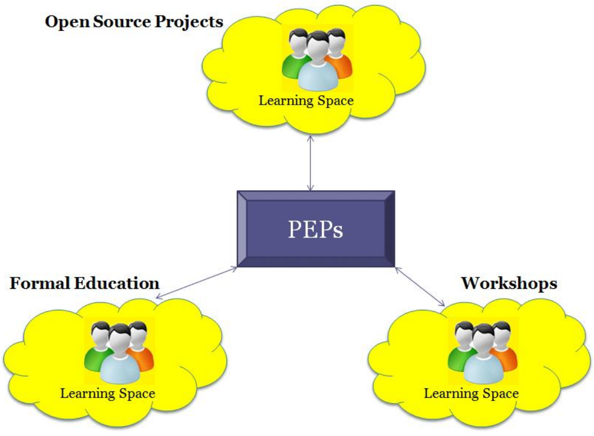


Figure 1. PEP system interacting with different learning spaces.

PEPs can import all kinds of learning activities: formal, informal or non formal learning, supporting LLLP. While importing the details from a learning space a PEP system asks the learner to authenticate. Then PEP imports learning activities from the given learning space. So the details obtained are authenticated and thus authenticity for evidences is provided. With the help of PEP system the user can create different views of her portfolio and can give access to others to view her portfolio.

Learners can participate in different learning spaces and can have different credentials for different spaces. It is hence problematic for the learners to remember all the credentials of different learning spaces. This problem is solved by introducing the OpenID authentication [Xia10]. OpenID is an URL, user-centered, open and decentralized standard for authenticating users. With the help of OpenID users do not have to remember the multiple usernames and passwords. In order to login into a system, a new user always has to register and also to different sites. Single Sign On (SSO) concept, means user logs into the system once and access to all the systems without giving login information again and again. As a solution to SSO, OpenID can simplify the user's operation process and reduce the resource provider overhead. i.e., OpenID has the single sign on procedure to reduce redundant, multiple accounts and passwords. Thus, the OpenID technology provides a secure and unified authentication mechanism to improve the anonymity of users.

PEPs support the interoperability of different learning spaces. They could show all types of artifacts that have been created by the learner, like for example assignments that they have completed and how educators or peers have assessed those, internships that they have completed, contributions to open source projects, certificates obtained, and other course interactions. In addition, PEPs could synchronize such information across the technical solutions provided by the different learning spaces. Thus PEPs can be thought as an implementation of composite portfolios.

In summary the benefits of PEPs include:

1. Connecting courses and programs to learner-created artifacts and to the underlying discourse and to synchronize them across technical solutions, thus allowing for re-usage and learning from what others have achieved;
2. Allowing for non-formal ways of recognition of learning outcomes within free / open learning by for example clearly showing learning outcomes and how those have been evaluated by peers or educators;
3. Providing a base for service providers to offer individual assessment and formal certification, as well as allowing service providers to build up their reputations;
4. Allowing for the connection of numerous education spaces and to take all kinds of information across such spaces. PEPs therefore could create a new model that allows learners and educators to carry education across institutions and other educational spaces; and therefore dovetails with lifelong learning.

4 Implementation

We have developed prototype tool support for PEPs, addressing the aforementioned requirements. From an architectural point of view, a PEP system is composed of the following components: PEP GUI, Learning space module, Learning space GUI, and OpenID module. Figure 2 shows the architectural design of the learning space environment. User interacts with the learning space through a PEP GUI. The learning space also provides a GUI for users to authenticate with their OpenID. If authentication is successful then the user is redirected to the PEP system and also the learning details from the learning space are saved into the PEP database.

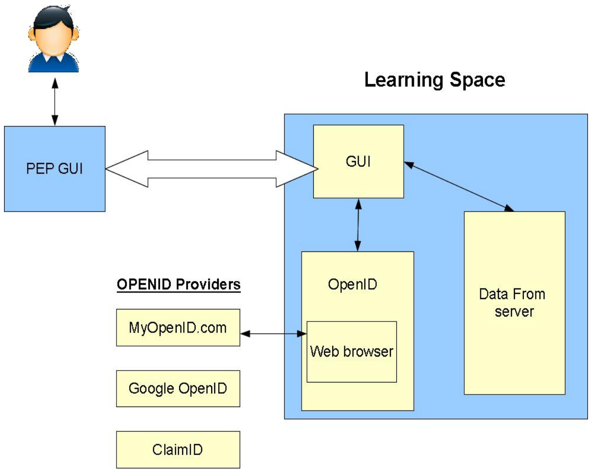


Figure 2. Architecture of PEP system.

The PEP engine collects user activities from learning spaces, designed using an extendible generic data model. The generic data model is developed in XML and typically contains information about the learner and the corresponding learning projects. The PEP engine implements an XML parser to parse data and saves the data in the PEP database.

PEP interacts with various learning spaces. In order to interact with learning spaces the user at least needs to have the following details:

- Address of learning space

The administrator of a learning space has to provide an URL which locates the learning space. The user has to provide this URL to the PEP system for PEP to be able to locate the learning space.

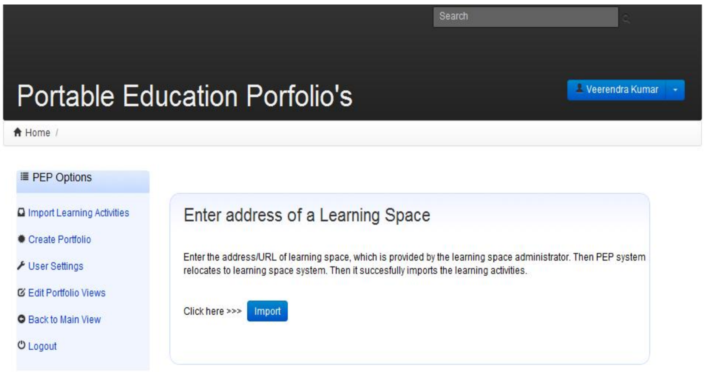
- Identify the user

The users have to associate OpenID with their learning space credentials. When the PEP system locates a learning space the user can authenticate with their OpenID. If the authentication process is successful the user gets recognized.

- Type of web service

The PEP system provides a sample data web service model. The administrator of a learning space has to provide the learning activity details of users in the same format of the model.

The PEP system provides an interface to users to specify the address of a learning space. The address is a Universal Resource Locator (URL) which is given by the administrator of that learning space to users. This is shown in Figure 3.



Portable Education Porfolio's

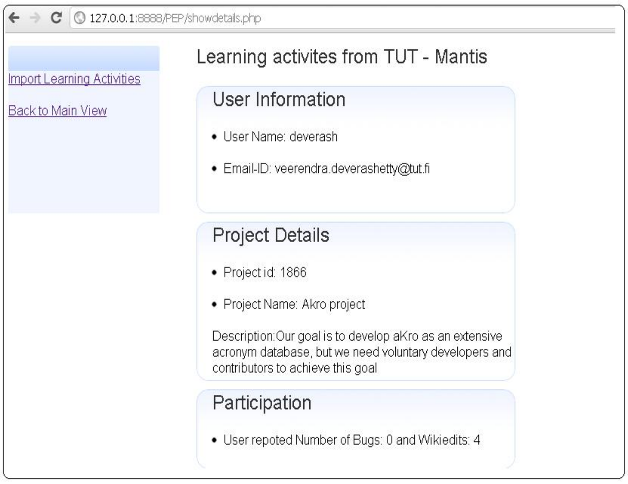
Enter address of a Learning Space

Enter the address/URL of learning space, which is provided by the learning space administrator. Then PEP system relocates to learning space system. Then it succesfully imports the learning activities.

Click here >>> Import

Figure 3. Interface to specify learning space location by user.

When the user successfully authenticates at a learning space the learning activity details of the user are saved to the PEP database. The PEP system provides a GUI named as PEP viewer which shows all the learning activities. This is depicted in Figure 4.


127.0.0.1:8888/PEP/showdetails.php
Import Learning Activities
Back to Main View

Learning activites from TUT - Mantis

User Information

- User Name: deverash
- Email-ID: veerendra.deverashetty@tut.fi

Project Details

- Project id: 1866
- Project Name: Akro project

Description:Our goal is to develop aKro as an extensive acronym database, but we need voluntary developers and contributors to achieve this goal

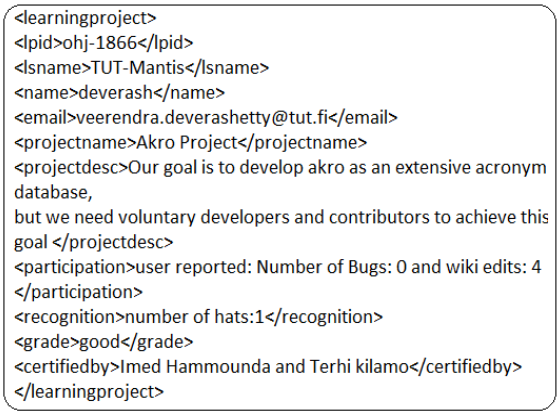
Participation

- User repoted Number of Bugs: 0 and Wikiedits: 4

Figure 4. PEP viewer showing learning details of user.

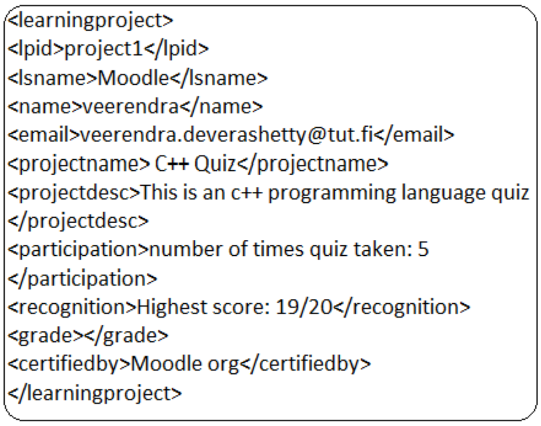
We have applied the developed PEP environment to integrate together three different learning spaces. The first is an open source project that uses Mantis bug tracking system [Mantis]. The second learning space is a formal education setting that uses Moodle [Moodle] as a course management system. The third learning space is the OpenSE [OpenSE] environment, which offers different kinds of learning projects such mentored internships and educational games.

Figure 5 and 6 show respectively sample data collected from Mantis and Moodle technical solutions. The two sample data is structured according to a well-defined data model.



```
<learningproject>
<lpid>ohj-1866</lpid>
<lsname>TUT-Mantis</lsname>
<name>deverash</name>
<email>veerendra.deverashetty@tut.fi</email>
<projectname>Akro Project</projectname>
<projectdesc>Our goal is to develop akro as an extensive acronym
database,
but we need voluntary developers and contributors to achieve this
goal </projectdesc>
<participation>user reported: Number of Bugs: 0 and wiki edits: 4
</participation>
<recognition>number of hats:1</recognition>
<grade>good</grade>
<certifiedby>Imed Hammounda and Terhi kilamo</certifiedby>
</learningproject>
```

Figure 5. Sample data collected from TUT-Mantis learning space.



```
<learningproject>
<lpid>project1</lpid>
<lsname>Moodle</lsname>
<name>veerendra</name>
<email>veerendra.deverashetty@tut.fi</email>
<projectname> C++ Quiz</projectname>
<projectdesc>This is an c++ programming language quiz
</projectdesc>
<participation>number of times quiz taken: 5
</participation>
<recognition>Highest score: 19/20</recognition>
<grade></grade>
<certifiedby>Moodle org</certifiedby>
</learningproject>
```

Figure 6. Sample data collected from Moodle.

5 Conclusions

In this paper, we have presented an approach to integrate heterogeneous learning spaces known as Portable Education Portfolios (PEPs). PEPs allow learners to collect different kinds of learning activities into unified education portfolios. The approach uses a generic data model that handles different types of web service platforms and ensures credibility of learning records.

Early experiences from a prototype environment for PEPs, that has been developed and used at Tampere University of Technology, have been promising. Using OpenID technology as authentication mechanism and XML-based data models, contributions from open source projects and formal course activities have been integrated into unified education portfolios. The adoption of the approach however required tuning of the technical solutions provided by the learning spaces. Finally, the PEP concept is not bound to learners. One could imagine the same concept applied to the context of teachers, with the goal in this case to record teaching, mentoring, and supervision history.

References

[Asp07] Aspin, D. N. and Chapman, J. D. Lifelong Learning Concepts and Conceptions. In: David N. Aspin, ed.: *Philosophical Perspectives on Lifelong Learning*, Springer. ISBN 1-4020-6192-7. 2007.

[Bha07] Bhattacharya, M. and Hartnett, M. E-portfolio Assessment in Higher education. In *proceeding of 37th ASEE/IEEE Frontiers in Education Conference*. Massey University, New Zealand. October 2007.

[Chr05] Chris Johnson and Cyprien Lomas. Design of the Learning Space: Learning and Design Principles. *EDUCAUSE Review*, vol. 40, no. 4 (July/August 2005): 16–28.

[Hen95] Henderson, J. An inside look at portfolio assessment. Paper presented at Interface 95, Lake Ozark, MO. 1995.

[Mantis] Mantis bug tracker. Available at http://www.mantisbt.org/. Last accessed August 2012.

[Mil05] Milman, N. B. Web-based digital teaching portfolios: fostering reflection and technology competence in preservice teacher education students. *Journal of Technology and Teacher Education*, Vol. 13, No. 3, pp. 373-397, 2005.

[Moodle] Moodle. Available at *http://docs.moodle.org/22/en/About_Moodle*. Last accessed August 2012.

[Obl06] Oblinge, D. G. *Learning spaces*, EDUCAUSE publishing, 2006.

[OpenSE] OpenSE website. Available at *http://opense.net/index.php*. Last accessed August 2012.

[Xia10] Xiangwu D. and Junyin, W. A scheme for confidentiality protection of OpenID authentication mechanism. In *proceedings of International Conference on on Computational Intelligence and Security*. Nanning, China. December 2010.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland