Open Source Software Opportunities and Challenges for Computer Science Instructors

Yves Lucet, Assistant Professor
Computer Science Department,
Okanagan University College
ylucet@ouc.bc.ca

Abstract

To increase students' adaptability and their ability to quickly assimilate new technologies, we introduced a variety of Open Source software (OSS) products in a third year visual programming course. We detail the rationale for using diverse OSS products, the challenges we have encountered, and give some suggestions to ease the instructor and students experience.

Introduction

Former graduate students from the computer science department at Okanagan University College (OUC) recognized that one great strength of the program is educating the students to the point they are able to learn and master new technologies on their own. To achieve this goal, course delivery usually involves a laboratory and a lecture component. The former illustrates the concepts and provide training on a variety of software tools, chosen based on their educational appropriateness, their market relevance, and their cost.

The increasing attention open source software (OSS) has been receiving [2], its current market share [1, 3], and the attention to the OSS development process [5, 6], led naturally to the consideration of that family of products to deliver better lab assignments. OSS has been used at OUC in courses like networking or operating system, where it has a relevant market share [4] or illustrates nicely the core concepts of the course.

While the advantages OSS boasts [6] - like licenses, privacy, and security - may not alone justify using OSS, the huge selection of software available - which has greatly mature in the last few years - allows instructors to find relevant tools for a large variety of courses. It also allows students to get training on how to learn software faster. Adaptation is a key factor in our field, so exposing students to a large variety of software improves their learning experience and increases their adaptation skills. Moreover, most current corporate environments are likely to include popular OSS like Linux, Apache, or mySQL and students already familiar with these have a competitive advantage in the job market.

While it is critical graduate students have experience in market leader software like MS Windows, Ms Office, Visual Studio .NET, Oracle, etc… The number of courses offered at OUC in the BCIS and BSc major in computer science allows the opportunity to additionally introduce other products to enhance the students' capability to adapt to new technology and to familiarize them with emerging technology.

To get such level of experience, OSS may be introduced in courses where it is on par with corporate software. We detail our introduction of OSS (Gnu/Linux, KDE, JBoss, and Eclipse) in our COSC 331 - Visual Programming II, our objectives, and make recommendations based on our experiment.

Experiment

The BCIS includes two courses on Visual Programming. COSC 131 – Visual Programming I introduces a graphical IDE. Visual studio and now Visual Studio .NET have been used for that purpose. The choice was based on market share, the ease of use of the IDE for first year students, and the educational value of the platform. In third year, students take COSC 331 – Visual Programming II to dive deeper into technological details through the exploration of APIs. In the past, the lab assignments used Ms Windows and Ms Visual Studio. In the Winter 2004, the decision was taken to use the J2EE environment in conjunction with OSS tools.

The objectives in introducing the J2EE environment and OSS were the followings.

- ***Market Share.*** To expose students to both the .NET and the J2EE platforms as it is likely both will exist in the future. .NET was presented in COSC 131. Our core programming courses COSC 111 – Programming I, and COSC 121 – Programming II are delivered using Java. So third year students have the background to follow a more advanced Java course focusing on the J2EE platform.
- ***Emerging technology***. OSS in general and the Linux platform in particular already has significant market share on niche markets like (web) servers [4]. Future predictions include a significant increase in market share so students will benefit from being already familiar with these technologies. Moreover, feedback from other courses indicated students felt they were unfamiliar with the Linux platform. So COSC 331 was to take the opportunity to fill that gap.
- ***Diversity.*** Introducing different platforms in the labs was to strengthen the students' capability to quickly adapt to new technologies. Microsoft and Oracle products were already exposed in other courses. So there was the opportunity to explore a different environment, and since Linux (and Unix) is introduced in courses focusing on networking or scripting, students had some experience with it. However, they were still not comfortable enough and COSC 331 was an opportunity to give them more exposure to such tools.
- ***Quality and availability of tools.*** OSS has matured enough to reach the level where students get a good experience in a teaching environment, or at least, as good as with proprietary software. The diversity of tools freely available also allowed the instructor to set up lab computers without being restricted by budgetary issues (or annoyed by activation/registration programs).

The following environment was selected for COSC 331.

- ***Linux*** was chosen to answer student feedback on their perceived lack of experience with it, and to take the opportunity to use it as a general-purpose platform. Mandrake Linux 9.1 (www.mandrakelinux.com) was the distribution selected based on its ease of use and the instructor previous experience.
- ***JBoss*** (www.jboss.org) was the J2EE server selected for its feature sets and its open source license. Tomcat was missing EJB implementation, and the instructor was uncertain Sun's reference implementation license could be used in an educational environment.
- ***Eclipse*** (www.eclipse.org) was chosen as the development tool. The instructor felt NetBeans was as mature as Eclipse but ultimately made his decision based on magazine reviews. At the time of the decision, both IDEs appeared to be able to deliver the course appropriately.
- ***Lomboz*** (www.lomboz.org), an Eclipse plug-in for J2EE, was selected for its documentation and the availability of tutorials.
- ***Open Office*** (www.openoffice.org) was used to produce assignment reports.
- Additional software like the Mozilla web browser (www.mozilla.org), the mySQL database (www.mysql.com), the openLDAP implementation (www.openldap.org), and the Ethereal network protocol analyzer (www.ethereal.com) were used in some of the assignments. They illustrated the richness and power of a Linux distribution, since most of these tools are already included with it.

All the software selected is available for download from the internet, providing students the opportunity to set up their home computer similarly to the lab and to get additional experience and flexibility in completing the assignments.

During the course of the term, the following issues were encountered:

- ***Lab setup.*** Since the Computer Services department at OUC does not support Linux, the instructor had to install and maintain all the software for the course. The main challenge resulted from a combination of lack of education from the students and inadequate cloning software. Specifically the version of Norton Ghost licensed for the lab did not support the journaling filesystem ext3 used by default with Linux Mandrake 9.1. So the non-journaling filesystem ext2 was used instead. However, computers in the lab use a multi-boot environment to maximize the use of the hardware and students shut down the computers improperly by turning the power off using the switch or even pulling the power cord. Approximately 3 to 5 computers out of 21 needed to be fixed every week because of file corruption issues.

  After a few weeks, the instructor converted all ext2 partitions to ext3 and updated the computers with rsync instead of Norton Ghost. No computer needed fixing after the conversion.
- ***The JBoss configuration*** was trickier than anticipated. The initial setup file generated errors on some services like CORBA (not used in the course), which created doubts in the students' mind about the maturity and the stability of JBoss. After more research, a stable configuration file was deployed. Students duplicating the lab configuration on their home computer still had issues originating from a combination of their lack of experience and a buggy configuration file. This issue was amplified by JBoss lack of free documentation.
- ***Students lack of experience with Linux*** created a few minor issues as anticipated. Mostly, student's perception of Linux was outdated. They looked for command line interfaces when a graphical interface was available, and compiled source code when pre-compiled binary packages were far easier to install.
- Some students took advantage of the multiplatform features of JBoss, Eclipse, Lomboz, and Open Office to complete their assignments using their home computer running MS Windows, thereby missing the opportunity to gain additional experience with Linux.

Conclusion and Suggestions

The following are suggestions for instructors wishing to use OSS in a similar setting.

- The main challenge is the lab setup particularly if your institution does not offer any support for Linux. One workaround is to use the same software under Ms Windows but students miss the opportunity to gain more experience with Linux. Should an instructor decide, like the author, to use Linux, be prepared to perform additional testing or to reset all computers in the lab a few times specially during the first few weeks. Afterwards, the lab should extremely stable.

  We found it impractical to have the software set up by students as part of their assignment. Students do benefit from the experience, but the resulting lab (as experienced in other courses) had very few computers set up similarly**.**
- To give students more experience administering Linux, all computers in the lab had the same username, password, and root password that were known to all students. If other courses had

been using the same system, different security arrangements would have been needed.

- Do show students how to setup, update, and install new products under Gnu/Linux, and any other specific procedures needed for the new software they are not familiar with.
- Take the opportunity to show OSS when it is on par with proprietary products like Open Office. Lead by example: if you find the tools adequate for your purposes, use them especially during in-class computer presentations. It will enlarge the student perception of the software world and invite them to explore new environments.
- Do not overdo it! The objective is to make student aware of mature tools that can help them in their future job, not to demonstrate that with enough time one can compile a full system. For a decade, it has been possible to compile Gnu/Linux but it is only in the past few years that one has been able to set it up quickly and easily.
- Keep up to date. New OSS reaches maturity all the time. Can you name an OSS product for most software categories?
- Our experiments revealed that students tend to assimilate OSS with the operating system Gnu/Linux. They have not yet realized the extent of the huge software stack available under open source licenses. Some are aware of Office productivity (Open Office.org), browser (Mozilla), and database (MySQL, PostgreSQL) products. However, students' initial reaction is to look for equivalent products running under MS-Windows. Moreover, students' perception of OSS was misunderstood e.g. asking to setup the PostgreSQL database in a lab resulted in some students downloading the source code to compile it instead of deploying the rpm package.

A key evaluation mechanism in most institutions is based on student satisfaction gathered through a student evaluation questionnaire. At the end of COSC 331, the students filled in a short survey. Two questions focused on the students' perception of the software selected in the lab. The following tables summarized the results.

Table 1: API the course should focus on

J2EE	56%
.NET	25%
Both	11%

Table 2: Operating System the lab should use

Linux	40%
Window	50%
Both	10%

Using the J2EE API was well received. The instability of the labs and (some) student reluctance to explore a new operating system explains the lower score for Linux. The continuous improvements of the Linux distributions should make it easier to use it on the desktop in the near future.

Other answers from the survey suggest to keep J2EE and Linux and to make sure all the lab setup is as stable as possible. Overcoming the support challenge will ensure the previously stated objectives are met, thus providing students with additional gains from the course.

References:

1. Tea leaves and Linux market share, Joe Barr, Aug 2003, http://www.newsforge.com/os/03/08/18/1516245.shtml?tid=2
2. Betting Big On Linux, Carolyn April, Apr 2004 http://story.news.yahoo.com/news?tmpl=story&cid=74&ncid=1209&e=10&u=/cmp/20040409/tc_cmp/18900632
3. Global IT firm predicts Linux will have 20% desktop market share by 2008, Chris Gulker, Aug 2003, http://www.newsforge.com/business/03/08/13/1424212.shtml
4. Netcraft Web Server Survey, April 2004, http://news.netcraft.com/archives/2004/04/01/april_2004_web_server_survey.html
5. Open Source Software Perspectives and Developments, Paul Dravis, Nov 2003 http://www-wds.worldbank.org/servlet/WDSContentServer/WDSP/IB/2004/03/24/000090341_20040324142743/Rendered/PDF/276800Open0Sou1re0nov0200301public1.pdf
   [Increased attention given to OSS]
6. Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers! David A. Wheeler, Dec 2003, http://www.dwheeler.com/oss_fs_why.html