

INTRODUCING PRESERVICE TEACHERS TO FREE AND OPEN SOURCE SOFTWARE: FINDINGS FROM A CASE STUDY

Ilias Karasavvidis
Department of Preschool Education
University of Thessaly
Greece

Abstract

Research suggests that one of the barriers to Information and Communication Technology (ICT) use in classrooms is teachers' lack of ICT skills and competencies. While for quite a long time ICT literacy meant familiarization with proprietary software, the growing importance of Free and Open Source Software (FOSS) has considerably changed the ICT landscape. The present paper examines undergraduate students' perspectives on FOSS after attending an introductory ICT course in which only such software was used. One hundred and one students from a preschool education department participated in the study. Two questionnaires were used for data collection. Data analysis indicated that (a) the students had no knowledge about FOSS concepts and applications and (b) GNU/Linux was considered more feature-rich and interesting compared to Microsoft Windows while Mozilla Firefox was also perceived as more feature-rich compared to Microsoft Internet Explorer. The paper is concluded with a discussion of the findings and implications for teacher training.

Teachers and ICT

Nowadays, most countries promote the use of ICT in education in an attempt to improve the quality of student learning. Consequently, ICT is a standard component of curricula in the western world. For at least three decades now, schools are being equipped with computers, educational software is being purchased, and teachers are being trained in ICT and how to use it in their practices. Regardless of the investment on infrastructure, teachers are not likely to use technology unless they have the knowledge, skills and attitudes to do so. Therefore, an important component of teacher training is related to the development of ICT competencies. The issue of technology skills for teachers needs to be addressed because research shows that teachers' lack of ICT knowledge and skills is one of the obstacles for the incorporation of ICT in their teaching (Hakkarainen et al., 2001; NCES, 2000; Pelgrum, 2001; Williams et al., 2000).

One might assume that even though practicing teachers lack technology skills, the young generation of prospective teachers are more prepared in terms of ICT. This is because students who are in their early 20s today grew up in a technologically

rich environment, had more exposure to ICT and are digital natives. Empirical evidence lends support to this notion. Teachers' skills and confidence in using technology appears to increase over the years (Condie et al., 2007) and teachers who are new in the profession (e.g. have 5–6 years of experience) are more confident in using computers compared to teachers who entered the profession a long time ago (Russell et al., 2003). On the other hand, research also suggests that even though young teachers possess more technology skills, there is a discrepancy between expected and actual technology skills (Albee, 2003). Thus, the level of technological preparedness of new teachers should not be taken for granted.

The issue of ICT literacy, skills, and competencies is important because it is a precondition for using ICT in educational practices. While research clearly suggests that technology training per se is not a sufficient condition (Wild, 1996), it is a necessary condition for without it the chances of teachers using ICT in their practices are minimal. Thus, the issue of preservice teachers' technology skills and preparedness needs to be explicitly addressed. In this paper we examine FOSS as a component of ICT-related teacher training.

FOSS and the Mature of ICT Literacy

The growing importance of FOSS, both in terms of software features and in terms of the underlying production model, has significantly changed the ICT landscape. FOSS is important for a number of reasons.

First, software-related freedoms. There are four kinds of freedom which characterize FOSS. These freedoms, which are defined in the licenses through which the software is distributed, grant users the freedom to run, copy, distribute, study, change, and improve the software (<http://www.fsf.org>; McGowan, 2005). Thus, the user can obtain the software without cost, has an unlimited number of licenses, may use the software for any purpose, may study and improve it as well as redistribute the improvements to the community so that others can also benefit. In the case of Proprietary Software (PS), these freedoms are simply inconceivable.

Second, features and qualities. As Chopra and Dexter (2008) note, a growing collection of powerful free software (e.g. Apache, BIND, Sendmail, GNU/Linux) have demonstrated their superiority to proprietary commercial software. More specifically, FOSS offers a number of advantages over PS including but not limited to reliability, security, performance, stability, cost, escape from vendor lock-up, scalability etc.

Third, market share, popularity, and adoption. FOSS' market share is on the rise both regarding operating system and applications software. In addition to the success in the corporate sector, many governments, local authorities, institutions,

and organizations also embrace FOSS. For example, the European Union heavily endorses the use of FOSS over PS in its attempt to become the most competitive knowledge economy by 2010. Moreover, a recent EU-funded study recommended avoiding PS in the educational systems of its member states (Ghosh, 2006).

Fourth, the production model. While on the surface FOSS is about software, in reality it involves much more than software. FOSS may initially have emerged as a model for software development, but it has gradually evolved into a phenomenon with far reaching effects (Lessig, 2005; O'Reilly, 2005; Raymond, 2001). FOSS is important because the collaborative ideas and principles which underlie its development can be applied to any collaboration which focuses on any kind of work or content (Schweik, 2007). For example, the contemporary Web 2.0 developments are based on principles and practices which originated with FOSS.

FOSS and Education

While, as a rule, ICT literacy entailed familiarization with PS, over the past few years FOSS has emerged as an important alternative in many respects. This has a bearing both on what ICT literacy involves and on the training of teachers. Teacher education departments aim to prepare teachers to use ICT in their practices. Given the growing impact of FOSS outlined above, the content of this training is a very fundamental issue.

It has been noted that PS has the potential to define the curriculum of e.g. Computer Science (CS) departments. Chopra and Dexter (2008) note that in such departments scientists often spend considerable time developing and teaching classes to train new users of commercial software. At the same time theoretical computer science loses its status in undergraduate curricula because it has little or no application in the workplace. Chopra and Dexter (2008) conclude that this need to train students in the successful commercial software programs available renders CS a vocational rather than scientific training.

One cannot fail to notice that once the pressure of PS is so high on CS curricula, the corresponding pressure exerted on social science departments, such as education departments, is even higher. This is because in education departments the emphasis on theoretical constructs is, by default, very limited while the emphasis on the development of software skills is enormous. As a consequence, most courses in education departments are skills courses. Of course, it comes as no surprise that these skills are almost exclusively related to PS. To a certain extent, this tendency to use PS was justified because of its relative dominance and market share position. However, over the past few years and in addition to its dominance in certain niches of the server market, FOSS has become an important contender on the desktop as well. Considering that the importance of FOSS is growing, we

argue that it is time to reexamine the content of ICT training in teacher training institutions.

Even though one can find several examples of studies reporting FOSS use in undergraduate courses (e.g. Carrington & Kim, 2003; Hernández-Leo et al., 2007; Raj & Kazemian, 2006), these primarily involve technical departments. It should be noted that introducing FOSS to technically sophisticated students with a CS background is less of a challenge compared to its introduction to education students. To the best of our knowledge, the issue of FOSS uptake has not been systematically explored with preservice education students, especially preschool and elementary school students.

Focus of the Study

The present paper draws on data from a larger research project on the use of FOSS in the ICT preparation of preservice preschool teachers. The paper examines students' familiarization with FOSS concepts and applications as well as their perceptions of 3 FOSS programs after attending an introductory ICT course in which FOSS was exclusively used. The study aimed to address the following research questions:

- What is students' familiarity with FOSS concepts and applications?
- What are students' perceptions of selected FOSS programs compared to PS ones in terms of features, ease to learn, ease to use, and general interest?

Method

Participants

One hundred one (101) students from a preschool education department participated in the study. This cohort of students, who were all female, had enrolled in an introductory, semester-long, ICT course which was compulsory for first year students that the author taught in his parent institution.

Course

The course aimed to introduce core ICT concepts as well as to render students skilled users of both operating system and applications software. Regarding the former, Ubuntu was the GNU/Linux distribution of choice considering that it is one of the most user-friendly distributions available. Regarding the latter, the course covered word processing and presentations (Writer and Impress from the OpenOffice.org suite respectively), web and mail clients (Mozilla Firefox &

Thunderbird), utilities (7Zip, InfraRecorder), and media player (VLC). The course involved weekly lectures and labs. It should be noted that because of the novelty of FOSS for students, lab attendance was compulsory.

Instruments & Measures

To the best of our knowledge, there exists no questionnaire which is suited for measuring familiarity with and attitudes to FOSS concepts and programs. Consequently, two questionnaires were specifically developed for the purposes of the study. The first instrument measured students' familiarity with common PS and FOSS applications as well as knowledge of main FOSS concepts. This instrument contained several questions on how skilled the students were with four general types of software: (a) operating system, (b) office, (c) Internet, and (d) multimedia. To measure familiarity with operating system software, 25 items on common tasks (ranging from simple ones such as copying files to more advanced ones such as customized software install) were used. The office category included questions about common software programs for FOSS and PS (e.g. Writer for FOSS, Word for PS). The Internet category included questions about FOSS and PS clients (e.g. Mozilla Firefox for FOSS, Microsoft Internet Explorer for PS). The multimedia category included questions about FOSS and PS programs which were related to graphics, audio, and video creation and processing (e.g. GIMP for FOSS, Adobe Photoshop for PS). A 5-point Likert scale ranging from unfamiliar (= 1) to very familiar (= 5) was used for all questions.

The second instrument measured students' views about various aspects of FOSS. This instrument included several items on FOSS programs. For the purposes of the present study, the dependent variables were 12 statements about 3 FOSS applications: GNU/Linux, Firefox, and OpenOffice.org. We focused mostly on those three applications because they were the ones which were thoroughly introduced in the context of the course. In each of the statements the students were asked to compare a FOSS program with its PS equivalent in terms of four dimensions: (a) features, (b) ease to learn, (c) ease to use, and (d) general interest. We chose to examine these four dimensions because they were deemed important. More specifically, the issue of features is critical for if students consider that the software lacks certain features (either because they are used to features that PS programs offer or simply because they tend to expect certain features), they will have very little incentive to use it. Regarding ease to learn, if students think that a program has a steep learning curve they will be less likely to turn to it in the future. The same holds for ease of use: if students find a program hard to use (compared to what they have been conditioned to use or what they expect) the program might not be a very appealing alternative to PS programs. Finally, the issue of general interest provides a general measure of how appealing the software is: the more interesting the program the more likely the students are to use it in the

future. The scale used for all 12 statements was a 3-point one, ranging from less (= 1) to more (= 3).

Procedure

The first questionnaire was administered at the beginning of the course. Depending on their reported familiarity with FOSS and PS applications, the students were assigned to four homogeneous competence groups ranging from novices to more advanced users. Next, the students attended a 13-week course which on a weekly basis included one 3-hour lecture on ICT concepts (e.g. data representation, computer architecture, algorithms, networks) and one 3-hr lab session. The lectures and labs were held on different days. Finally, the second questionnaire was administered at the end of the course.

Analysis

Regarding the first question, composite scores were computed for operating system, office, Internet, and multimedia programs for both FOSS and PS. Means and standard deviations were computed for each composite variable and the Wilcoxon signed ranks test was used to detect any differences in familiarity between FOSS and PS applications. To examine the levels of familiarity with FOSS concepts as well as the importance attributed by the students to the four fundamental FOSS freedoms, means and standard deviations were computed. Regarding the second question, student responses on the 12 statements were initially recoded into two categories, one for less or same and another for more. The chi-square test was subsequently used to compare the frequencies obtained so as to determine whether students systematically favored FOSS applications over PS ones in the dimensions examined.

Results

Familiarization with software applications

Operating system software. Students reported familiarization with the GNU/Linux operating system was virtually non existent: $M = 1.07$, $SD = 0.48$ while they reported that their familiarization with the proprietary Microsoft Windows operating system was moderate: $M = 2.8$, $SD = 1.28$. While only 12.9% of the students stated that they were completely unfamiliar with Microsoft Windows, 97.6% of the students reported that they were unfamiliar with GNU/Linux. The Wilcoxon signed ranks test confirmed the obvious, i.e. that the students were more familiar with the PS operating system at a statistically significant level ($z = -7.0406$, $p = .000$).

Office software. The students were asked to rate their familiarity with the component applications of both OpenOffice.org and Microsoft Office suites. The

students reported that their familiarity with OpenOffice.org was minimal ($M = 1.16$, $SD = 0.45$). On the other hand, students' reported familiarity level with Microsoft Office was moderate ($M = 2.57$, $SD = .20$). Eighty-five percent of the students stated that they had no knowledge about any of the OpenOffice.org suite applications (Writer, Calc, Impress, Base) while only 18.8% of the students reported being unfamiliar with any of the Microsoft Office suite applications. As expected, the Wilcoxon signed ranks test indicated that students were systematically more familiar with the Microsoft Office suite ($z = -7.195$, $p = 0.000$).

Internet software. The students reported virtually no knowledge about FOSS Internet applications (Firefox, Thunderbird, etc.) ($M = 1.35$, $SD = 0.68$) while their knowledge of PS Internet applications was low ($M = 2.1$, $SD = 0.87$). Again, this difference was found to be statistically significant in favor of PS ($z = -6.580$, $p = .000$).

Multimedia software. This was the only software category where the students reported having very little knowledge about the component applications (e.g. GIMP, Adobe Photoshop, VirtualDub, Adobe Premier, etc.). More specifically, as far as the FOSS applications are concerned, students' reported familiarity was very low ($M = 1.22$, $SD = 0.41$). Regarding PS applications, the students also reported low levels of familiarization ($M = 1.27$, $SD = 0.49$). A comparison of the two using the Wilcoxon signed ranks test indicated no significant differences ($z = -1.00$, $p = .454$).

Familiarization with FOSS concepts

Students' familiarization with the main FOSS concepts is presented in Table 1. As can be seen from the table, the students were essentially unfamiliar with FOSS concepts. It should be noted that more than 95% of the students reported having no knowledge whatsoever about what FOSS stands for. On the other hand, the examination of the importance that the students attributed to using the software for any purpose, adapting the software to one's needs, zero cost, and legal licenses indicated that the students were not much concerned with these issues. It should be emphasized that these four features constitute the four main freedoms which characterize FOSS. While some students did state that they were concerned e.g. about possessing a legal license for the software they use, as a group this cohort of students did not seem to be very concerned about the main FOSS freedoms.

Table 1: Students' reported familiarization with main FOSS concepts and importance attributed to the fundamental freedoms

	Variable	M	SD
Familiarization^a	Software License	1.23	0.81
	Free Software (FS)	1.16	0.75
	Open Source Software (OSS)	1.09	0.50
	Free & Open Source Software (FOSS)	1.08	0.44
	FOSS Cost	1.05	0.30
	How to obtain FOSS	1.11	0.49
	use software for any purpose	2.93	1.00
Importance^b	adapt software to one's needs	3.05	1.12
	software cost	3.02	1.16
	legal licence	2.96	1.15

a. 5-point Likert scale (1 = unfamiliar, 5 = very familiar)

b. 5-point Likert scale (1 = unimportant, 5 = very important)

Comparisons between FOSS & PS

Students' perceptions of 3 FOSS programs as compared to the corresponding PS programs are presented in Table 2.

Office suites. As Table 2 indicates, no statistically significant differences emerged between OpenOffice.org (OO) and Microsoft Office (MSO) in any of the four dimensions compared. While more students found OO to have more features compared to MSO (31 vs. 22) this difference was not statistically significant. On the other hand, none of the students found OO to be more interesting compared to MSO (55 vs. 0). Even though the inferential statistic could not be computed, it is obvious that the difference is statistically significant.

Web clients. When it comes to browser comparison, the analysis indicated that the students considered that Mozilla's web client had more features compared to MSIE. On the other hand, they did not find Firefox to be easier to learn or use compared to MSIE. Interestingly enough, Firefox was rated as being more interesting than MSIE but the chi-square value was not statistically significant (although it approached significance: $p = .052$).

Operating systems. The most noticeable differences were found in comparing the two operating systems: GNU/Linux and Microsoft Windows (MSW). Regarding features, the students found GNU/Linux to be more feature-rich compared to MSW. What is more, the results indicated that the GNU/Linux operating system

captivated student interest significantly more than MSW. On the other hand, no statistically significant differences emerged with respect to the ease of learning GNU/Linux compared to MSW. Finally, the students found GNU/Linux to be more difficult to use compared to MSW.

Table 2: Students' perceptions of FOSS vs. PS programs

SW Category	Measure	Less-same	More	Chi-Square	df	p
Open Office.org	Features	22	31	1.528	1	0.272
	Ease to learn	29	20	1.653	1	0.253
	Ease to use	26	17	1.884	1	0.222
vs.						
Microsoft Office	Interest	55	0	n.a.		
Mozilla Firefox	Features	15	37	9.308	1	0.003
	Ease to learn	22	29	0.961	1	0.401
	Ease to use	19	30	2.469	1	0.152
vs.						
Microsoft Internet Explorer	Interest	22	38	4.267	1	0.052
GNU/Linux	Features	10	43	20.547	1	0.000
	Ease to learn	26	17	1.884	1	0.222
	Ease to use	31	16	4.787	1	0.040
vs.						
Microsoft Windows	Interest	14	42	14.000	1	0.000

Discussion

This paper examined preservice teachers' (a) familiarity with FOSS concepts and programs and (b) perceptions of FOSS programs as compared to PS ones. With respect to the first research question, data analysis indicated that the students were largely unfamiliar with both FOSS concepts and programs. Moreover, the students were systematically more familiar with PS programs compared to FOSS ones: the students' self-reported familiarity with PS programs outperformed the corresponding one with FOSS programs — with the exception of multimedia creation and processing. Needless to say this finding comes as no surprise considering how widespread PS is. In addition to being largely ignorant about the main FOSS concepts, the students did not appear to value the fundamental FOSS freedoms very much.

With respect to the second question, the analysis did not show any significant differences between the two office suites compared. It is interesting to note, however, that all students found Microsoft Office to be more interesting than OpenOffice.org. This finding is understandable because the version of OpenOffice.org used in the course, i.e. 2.0, was not as polished as Microsoft Office in terms of the interface. Students' perceptions of the browser comparison indicated that they were enthusiastic about Firefox which they found to be more feature-rich compared to Microsoft Internet Explorer. The students were excited by tabbed-browsing, bookmark management, and most importantly, browser-plugins. Regarding students' perceptions about the operating system comparison, the analysis indicated that the students found GNU/Linux to be more interesting and feature-rich compared to Microsoft Windows. The students were enthusiastic about Live-CDs, different GUIs, multiple desktops, and software repositories-package management system among others. On the other hand, the students found GNU/Linux to be easier to use compared to Microsoft Windows. While the students did not find GNU/Linux to be a geek-only operating system, they still appeared to be concerned about the ease of use. Thus, the students did recognize the potential of GNU/Linux but considered that Microsoft Windows was easier to use.

The findings of the present study suggest that while the students were utterly unfamiliar with FOSS concepts and programs, their ratings of FOSS programs in comparison with PS ones for the most part favored FOSS programs. The students appreciated the features of GNU/Linux and Mozilla Firefox, and expressed much interest in GNU/Linux. At the same time, they found Microsoft Office more interesting than OpenOffice.org and Microsoft Windows easier to use than GNU/Linux. Overall, students' responses to FOSS programs as a result of the course introduction greatly exceeded our expectations. The findings are very promising because FOSS — and especially GNU/Linux — is surrounded by several myths, e.g. it is notoriously difficult to install, master, and use effectively. Using students' responses as a criterion, the present study indicated that the introduction to FOSS in the context of an introductory ICT course may lead to successful outcomes. Consequently, FOSS might be effectively used to foster prospective preschool teachers' ICT literacy skills.

In conclusion, given the growing importance of FOSS both in terms of software quality and in terms of a new model of social practice (e.g. wikinomics) we argue that the training of student teachers in the use of FOSS is imperative. As the present study suggested, a systematic approach to FOSS might yield positive experiences and enrich students' ICT skills and perspectives. Especially when it comes to preservice teachers, teacher ICT training should expand to include FOSS.

References

- Albee, J. J. (2003). A study of preservice elementary teachers' technology skill preparedness and examples of how it can be increased. *Journal of Technology and Teacher Education*, 11(1), 53–72.
- Carrington, D., & Kim, S-K. (2003). Teaching software design with open source software. *Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference*. Boulder, CO.
- Chopra, S., & Dexter, S. D. (2008). *Decoding liberation: The promise of free and open source software*. London: Routledge.
- Condie, R., Munro, B., Seagraves, L., & Kenesson, S. (2007). *The impact of ICT in schools — A landscape review*. BECTA.
- Ghosh, R. A. (2006). *Economic impact of FLOSS on innovation and competitiveness of the EU ICT sector*. Final Report. MERIT. Retrieved February 28, 2008, from <http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf>
- Granger, C. A., Morbey, M. L., Lotherington, H., Owston, R. D., & Wideman, H. H. (2002). Factors contributing to teachers' successful implementation of ICT. *Journal of Computer Assisted Learning*, 18, 480–488.
- Hakkarainen, K., Muukonen, H., Lipponen, L., Illomaki, L., Rahikainen, M., & Lehtinen, E. (2001). Teachers information and communication technology (ICT) skills and practices of using ICT. *Journal of Technology and Teacher Education*, 9(2), 181–197.
- Hernández-Leo, D., Bote-Lorenzo, M. L., Asensio-Pérez, J. I., Gómez-Sánchez, E., Villascclaras-Fernández, E. D., Jorrín-Abellán, I. M., & Dimitriadis, Y.A. (2007). Free- and open-source software for a course on network management: Authoring and enactment of scripts based on collaborative learning strategies. *IEEE Transactions on Education*, 50(4), 292–301.
- Lessig, L. (2005). Open code and open societies. In J. Feller, B. Fitzgerald, S. A. Hissam, & K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software* (pp. 349–360). Cambridge, MA: The MIT Press.
- McGowan, D. (2005). Legal aspects of free and open source software. In J. Feller, B. Fitzgerald, S. A. Hissam, & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 361–391). Cambridge, MA: The MIT Press.
- O'Reilly, T. (2005). The open source paradigm shift. In J. Feller, B. Fitzgerald, S. A. Hissam, & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 461–481). Cambridge, MA: The MIT Press.
- Pelgrum, W. J. (2001). Obstacles to the integration of ICT in education: Results from a worldwide educational assessment. *Computers & Education*, 37, 163–178.
- Raj, R. J., & Kazemian, R. (2006). Using open source software in computer science courses. *Proceedings of the 36th ASEE/IEEE Frontiers in Education Conference*. San Diego, CA.

- Raymond, E. S. (2001). *The cathedral and the bazaar. Musings on Linux and open source by an accidental revolutionary*. Sebastopol, CA: O'Reilly & Associates.
- Rusell, M., Bebell, D., O'Dwyer, L., & O'Connor, K. (2003). Examining teacher technology use: Implications for preservice and inservice teacher preparation. *Journal of Teacher Education*, 54(4), 297–310.
- Schweik, C. M. (2007). Free/open source software as a framework for establishing commons in science. In C. Hess & E. Ostrom (Eds.), *Understanding knowledge as a commons: From theory to practice* (pp. 277–309). Cambridge, MA: The MIT Press.
- Wild, M. (1996). Technology refusal: Rationalizing the failure of student and beginning teachers to use computers. *British Journal of Educational Technology*, 27, 134–143.
- Williams, D., Coles, L., Wilson, K., Richardson, A., & Tuson, J. (2000). Teachers and ICT: Current use and future needs. *British Journal of Educational Technology*, 31(4), 307–320.