It's Time to Consider Open Source Software

By Jay Pfaffman

A Crazy Man's Dream

In 1985 Richard Stallman, a computer programmer, released "The GNU Manifesto" (Stallman, 1985) in which he proclaimed a golden rule: One must share computer programs. Software vendors required him to agree to license agreements that forbade sharing programs with others, but he refused to "break solidarity" with other computer users whom he assumed also wanted to use free software.

"Free software gives everyone the freedom to run, study and change, and redistribute software. It is these freedoms, not the price, that is important about free software."

Many people are surprised to hear that Stallman's free software dream has been realized. This paper makes the case that using only free software has considerable economic, technical, political, pedagogical, and moral advantages – and surprisingly few frustrations. If you are a teacher, and especially if you train teachers, you should be aware that there are Free/Open Source Software (F/OSS) applications for common classroom uses.

F/OSS is the foundation of the Internet. The BIND name server that maps domain names to Internet Protocol (IP) numbers, the Apache web server that serves most web sites, the Linux kernel that drives Google and Amazon, and the MediaWiki software that powers Wikipedia are all free for anyone to use, change, and redistribute. These examples demonstrate that F/OSS tools can be the best solution in certain situations, but for those working in K-12 classrooms these server-based examples are easily dismissed as esoteric or unimportant. In recent years, however, open source software developers have released a wide range of end-user applications that can replace most of the applications currently used in K-12 classrooms.

The Power of the Source

A key aspect of F/OSS is the availability of the *source code* – the human-readable text files used to create the program. Accessing the source code allows anyone to examine the program to see how it works, fix bugs, or change it to suit personal needs. Like freedom of speech, one does not need to use source code to benefit from it. Free software gives everyone the freedom to run, study, change, and redistribute software. It is these freedoms, not the price, that is important about free software. Free software advocates make the distinction between free, as in speech, as opposed to free, as in beer. Though many people would gladly accept a free beer, it is not one of the fundamental principles of democracy.

Property Rights Turned Upside Down

Those of us who have long been familiar with open source software need to understand that the concept of free software is foreign to many

people. The conventional notion of property rights is that I have the right to exclude you from using something that belongs to me. Open source software is based on the opposite belief – that *everyone* has the right to distribute and *no one* has the right to exclude further distribution. This paradigm shift can be difficult to make. For three years, F/OSS has been a central part of my course on using computers in the classroom and I give conference presentations about F/OSS at least twice a year. I was getting good at it. Recently I gave a conference presentation about the benefits of F/OSS for educators – how all teachers and students could use these tools and that they were free and would remain so. I distributed copies of TheOpenCD (2006) and talked about the F/OSS programs that it includes. Near the end of the hour-long presentation a participant raised her hand and asked "So I can use this software for free?" Even after an hour, F/OSS still did not quite make sense to her.

On the Annoyances of Proprietary Software

Proprietary software is inconvenient. It is inconvenient to purchase additional licenses when you buy new machines. It is inconvenient to negotiate a new license agreement each year. It is inconvenient to support multiple versions of a package for machines purchased at different times. It is inconvenient to be faced with an ethical dilemma when a friend or colleague asks to illegally copy the proprietary software that you support. It is inconvenient for students not to have the same software at home and at school. It is inconvenient to decide what to do when a vendor starts charging for a proprietary program that was previously free. It is inconvenient to perform an audit to document that all software on every computer has a valid license (Acohido, 2002; Cave, 2001). Thanks to the work of Stallman and thousands of others, these inconveniences are now avoidable.

Understanding Open Source Software

This section addresses questions and concerns that I hear from students in my instructional technology courses as well as from teachers and technology leaders that I meet in conference presentations.

Myth: You Get What You Pay For

With traditionally distributed software you pay for packaging; you pay for manuals; you pay for marketing; you pay for support, regardless of whether you receive it; you pay for programmers; you pay dividends to shareholders. You may get what you pay for, but someone else decides the cost. With open source software you can still pay for those things, but you control who and how much to pay.

Myth: F/OSS Software is Created by Amateurs and Must be Inferior

Two fallacies follow from this assumption. First, that open source applications are not developed by professional programmers. Second, that amateur programmers produce inferior work.

"Proprietary software is inconvenient."

Many F/OSS projects have teams of professional programmers. RedHat software distributes a version of GNU/Linux, an open source operating system that for many people obviates the need for Microsoft Windows (BECTA, 2005; Zetter, 2002). Other projects, like Firefox and OpenOffice.org, are based on commercially developed code (StarOffice and Netscape Navigator, respectively) that was subsequently released as open source (Raymond, 1997; Baker, 2004). Both of these projects have improved dramatically since becoming F/OSS.

Economists are starting to recognize the impact of ProAms –people who do high quality work in a field other than the one that earns them their money (Leadbeater & Miller, 2004). Amateur astronomers regularly make new discoveries. Amateur musicians and songwriters generate money for coffee houses and bars. Einstein wrote several of his most famous papers while he worked in a patent office. So it is with software. Many people develop software because they enjoy it. Some do it for the respect they gain in online communities that form around the creation of open source applications.

Myth: With F/OSS I Cannot Get Support

This myth follows from the assumption that when you pay for something you have certain rights and expectations. Though this may be true for physical objects, most software licenses exclude you from such rights. For example, Microsoft's End User License Agreement (EULA) explicitly states that there is no warranty after ninety days (Zymaris, 2003). Most often when one needs help to learn to use or configure a piece of proprietary commercial software, the best option is a third party. The first place most teachers go for support is to experts in the building. What happens, then, when the person being asked for help has a different version of the program in question than the person asking the question? F/OSS makes this situation unnecessary since there are no license restrictions keeping everyone from upgrading to the same version simultaneously.

Myth: Moving to F/OSS Will Require Retraining and Re-learning

Since Apple published user interface standards (Apple, 1987), many of which were quickly copied by Microsoft, most computer applications work similarly. For the most common operations, it makes little difference whether one uses ooWriter, AbiWord, Google Documents, or any of several versions of Microsoft Word.

"There are dramatic advantages to arming students with tools that assurethem access wherever they go."

People are often reluctant to try new computer programs, though most users find only subtle differences between one program and another. In the course of giving conference presentations about F/OSS, two computer coordinators shared stories of upgrading some users' Microsoft Office suite with OpenOffice.org., leaving Microsoft Office icons as the means to start Open Office.org. In both cases, most users failed to notice that they were no longer using Microsoft Office.

Myth: Students Need to Learn the Standard Applications

Schools have a responsibility to give students the skills they need to succeed. By the time high school students get to the job market, today's applications will be antiquated. Students need to know how to use word processors to communicate and spreadsheets to explore numbers and graphs. Their technical skills should transcend the particular idiosyncrasies of the applications.

Why F/OSS is Important for Educators

The previous sections have discussed the availability of F/OSS and how switching to it is less difficult than many people assume. This section goes a step further to suggest that educators and especially teacher educators should use F/OSS whenever possible and that the freedom offered by F/OSS can outweigh shortcomings in the software itself.

Educators Pay for Software – Twice

Unlike diamonds, software applications are more valuable when more people have them. This is among the reasons that Microsoft Office is so popular. Everyone uses it because everyone uses it. When enough of your friends upgrade to a newer version that makes files that you can't read, you upgrade too.

Training teachers and students to use a piece of software makes that software more valuable. Vendors know this. Business sense, not altruism, is what drives deep discounts on software for education. I once spoke to a vendor of an online grade book who, upon learning that I train teachers, was very interested in my using it in my classes.

"What does it cost?" I asked.

"It will cost you nothing. You can use it for free for as long as you like."

"And once I addict my students to your software," I asked pointedly, afraid that I was being rude, "what will it cost them?"

The vendor became excited. "That's *exactly* what we were talking about in our last sales meeting!"

When technology leaders train teachers and students to use proprietary software, it obligates those teachers and students to buy or steal that software or to have wasted their time on the training. A $500 Dell computer costs another $150 (an additional 30%) with Microsoft Office Basic; the version for students and teachers is available for about $100 (20%). When one considers how many students and teachers buy Microsoft Office because their school has standardized on it the costs are considerable. A medium-sized school district comprises about 20,000 households with children; if 10 percent of them spend $100 to help their students do better in school by buying the software that the school is using, the district has spent $20,000 of its constituents' money. Whether schools should be effectively taxing its constituents in this way is an issue that deserves more attention.

Training Teachers on Tools They Do Not Have

Though it is a safe bet that students will have the skills they need to access certain proprietary tools like an office suite wherever they go (and that these skills are fairly interchangeable between competing applications), software for photo editing, desktop publishing, web editing, statistics, and symbolic mathematics are not so widely available. There are dramatic advantages to arming students with tools that assure them access wherever they go. Teacher educators cannot ensure that teachers will have the hardware that they need to integrate technology into their teaching, but we can provide them with tools that they will have the right to use in perpetuity. Teachers spend their own money for supplies like pencils and paper; encouraging them to spend money on software when free alternatives are available is unnecessary.

On the Allure of Free Proprietary Tools

Some proprietary applications have "free" versions with fewer features, allow "free" use for education, provide a "free" limited use version, or are simply available free of charge, but they

are not Free. F/OSS licenses guarantee that the software will be freely available forever. When software is free, but not Free, things can change. My university adopted a file transfer program that was free for educational use, but the company later changed its policy and started charging for the software. A free-of-cost program that I once used to control CD changers became unavailable when the company producing it was purchased by Microsoft. Some free-of-charge programs are likely to be free forever. Adobe Reader, for example, will likely remain free since its large user base is what makes Adobe's PDF-producing applications marketable. Even these programs, however, restrict your rights to redistribute the software.

Show Me the Money: Examples of High Quality F/OSS

This section includes examples of F/OSS that all technology educators should know about and make known to their students. Most of these can be used as drop-in replacements for their proprietary alternatives. Table 1 provides a synopsis of these applications and their platforms, though most are cross-platform.

Productivity Applications

Perhaps the most important applications are those that allow students and teachers to create papers, pictures, and sounds and manipulate numbers. The Microsoft Office suite is the *de facto* standard. In the past few years, however, OpenOffice.org's suite has matured considerably. The feature that it lacks that is most likely to be a problem is a grammar checker; other than that for most tasks, it is quite capable. Computerworld (2005) suggests that it will be easier for many users to adopt OpenOffice.org than the forthcoming Office 12. For those with older slower computers or the need for a grammar checker, AbiWord, is a good option. It lacks some features of OpenOffice.org's ooWriter, but for most needs, it is quite sufficient.

The GNU Image Manipulation Program (the GIMP) is a very capable photo editing package. It has similar features to Photoshop, but a different menu structure. For those who already know how Photoshop has things arranged, another set of programmers has re-arranged the GIMP's menus in a program they call GIMPShop. Inkscape is a vector-based drawing program that is not yet as full-featured as its proprietary counterparts, but is quite capable for a wide range of tasks. For those interested in 3D modeling, rendering, and animation, Blender offers many of the features provided by very expensive packages. Since such a specialized program is likely to be used by only a few students, it can be an attractive option. Tuxpaint is a paint program with stamps and sounds for younger kids, somewhat similar to KidPix. It has an easy-to-use interface and saves files without the bother of file names.

Nvu is an HTML editor that is based on code from Mozilla. One of its best features is a simple tool for managing a remote web site. Nvu is more than adequate for the kinds of web pages that most K–12 teachers and students need to create, Like other HTML editors, it has multiple views, allowing users to edit HTML by hand or use word processor-like commands to effect the same results.

Scribus is a desktop publishing package that is quite mature. In addition to the expected desktop publishing features, Scribus can also create animated and interactive PDFs. PDFCreator installs as a printer driver for those who need to create simple PDFs, allowing any Windows program that can print to create PDFs.Audacity is an easy to use digital audio editing program that has become popular for podcasting, though it is also capable of doing multi-track digital recording work.

"Teachers spend their own money for supplies like pencils and paper; encouraging them to spend money on software when free alternatives are available is unnecessary."

Freemind is a Java-based open source concept mapping tool. It does not offer as many pictures as its proprietary counterpart, but does allow publishing maps to a web page. CmapTools is another free concept mapping package. It is not Free or open source, but is distributed by a university rather than a commercial entity; it could, therefore, become unavailable in the future.

Internet Applications

Firefox is a web browser that includes features not found in Microsoft's Internet Explorer (IE). It is also generally recognized to be less susceptible to viruses than is IE (Krebs, 2007). Firefox has an easy-to-use mechanism for installing add-ons (or extensions) and the open source community has responded by developing hundreds of them. Adblock Plus, for example, will remove ads from web pages, reducing download time and distractions.

Gaim is a multi-protocol instant messaging (IM) client. It can connect to all of the most popular IM networks, obviating the need to install separate clients for each. Adium uses some of the code from Gaim for a similar application

Name	Description	Commercial example	Platform
	Internet Applications		
Firefox	Web browser	Internet Explorer, Opera	Cross
Gaim	Multi-protocol instant Messager client.	Yahoo!, AIM,	Linux, Windows
Adium	Multi-protocol instant Messager client.	Yahoo!, AIM,	Mac
FileZilla	FTP, SFTP client and server	SmartFTP, WS_FTP	Windows
WinSCP	FTP, SFTP, SCP client	SmartFTP, WS_FTP	Windows
Fugu	FTP, SFTP, SCP client	Fetch	Mac
Cyberduck	FTP, SFTP, SCP client with directory synchronization	Fetch	Mac
	Productivity		
AbiWord	Word Processor	Microsoft Word	Cross
OpenOffice.org	Office Suite	Microsoft Office, StarOffice	Cross
GIMP	Photo Editing software	Photoshop	Cross
GIMPShop	Photo Editing software with Photoshop's menu structure	Photoshop	Cross
Inkscape	vector graphic editor	CorelDraw, Adobe Illustrator	Cross
Audacity	Sound editor	Adobe Audition, Sony Sound Forge	Cross
Blender	3D Modeling and animation	Maya	Cross
Nvu	HTML editor	FrontPage, Dreamweaver	Cross
Scribus	Desktop publishing	Pagemaker, InDesign, QuarkXPress	Cross
PortableApps	Installs applications and their settings on a USB drive		Windows
PDFCreator	PDF Creation	Adobe Acrobat	Windows
Tuxpaint	Drawing program for kids ages 3-12	Kid Pix	Cross
FreeSMUG Portable Apps	Installs applications and their settings on a USB drive		Mac
CmapTools	Concept Mapping	Inspiration	Cross
FreeMind	Concept Mapping	Inspiration	Cross
	Math		
Axiom	Symbolic & Algebraic calculator	Mathematica, Maple	Windows, Linux
Sage	Symbolic & Algebraic calculator	Mathematica, Maple	Cross
Maxima	computer algebra system	Mathematica, Maple	Windows, Linux
Yacas	Computer algebra system, will run in a web browser	Mathematica, Maple	Cross
	Science		
Celestia	Astronomy. Allows travel about the Universe	Deepsky	Cross
Stellarium	Planetarium	Deepsky	Cross

for the Macintosh. Schools that would like to have teachers or students use IM but want control over who can use the service should investigate Jabber, an open IM protocol with cross platform server applications. This software enables schools to provide IM service with complete control over who can participate.

WinSCP and FileZilla are FTP and SCP clients for Windows. Unlike many other free FTP clients, they have no license reminders to annoy users or expirations that require re-installation. Cyberduck is an FTP/SFTP client for the Mac that will also synchronize local and remote folders. Fugu is an SCP front end for the Mac with drag-and-drop support and the ability to upload entire folders.

Many of the applications mentioned thus far are also available as *portable applications.* These are special versions designed to run from a USB drive. This enables users to carry their data *and*

their applications and their settings with them. For example, Portable Firefox allows users not only to use Firefox on computers that do not have Firefox installed, but also to have their bookmarks, add-ons, and other settings with them.

Content-Specific Applications

Axiom and Maxima are computer algebra systems that will do symbolic calculations and create graphs. Yacas is another such system that can also be run in a web browser. Sage will run in a web browser and provides a bootable CD that will set up one computer as a server, allowing other computers in the school to use Sage via a web browser.

For those interested in teaching or learning astronomy, Celestia and Stellarium are exciting programs. Stellarium is a computer-based planetarium that will show the stars just as they are in the sky. Celestia will simulate moving around the galaxy, showing planets and their orbits.

Server-Based Applications

Course management systems like WebCT and Blackboard are ubiquitous in colleges of education, but expensive to buy and cumbersome to maintain. Moodle provides an alternative that is easier and requires less training for the teacher, the student, and the technology support department (Pfaffman, 2005). Though Moodle is designed as a platform for online courses, it also makes it easy for K–12 teachers to create web sites for their courses, letting students, parents, and other teachers know what is going on in their classes (Perkins & Pfaffman, 2006). Moodle is a web-based application, so teachers can update their site anywhere without installing or configuring HTML editing software. Because Moodle is not only free but also requires little support, my university is able to provide Moodle sites for any school – or teacher – who asks. Students in my class who create course content in Moodle know that they can move it to their own Moodle server to use with their own students. Because Moodle is easy to run and will work on Windows, Mac OS X, and GNU/Linux servers, several of my students have introduced Moodle to their schools and had their school or district adopt it. Sakai, developed by a consortium of schools including Stanford and the Massachusetts Institute of Technology, is another course management system that is somewhat more powerful but considerably more difficult to install and configure.

Other interesting server-based applications include Koha and OpenBiblio, integrated library systems capable of managing any K-12 school's collection. Squirrelmail is a very usable web-based mail system.

Conclusion

Because it is the norm in most schools, businesses, and homes, many of the costs of proprietary software are difficult to see. There are now alternatives to the most commonly used applications in schools. When these open source alternatives are nearly equal to – or better than –their proprietary competitors, the significant advantages of F/OSS make them the better choice. When making the decision to license a proprietary product schools need to carefully consider whether their needs may be better met by open source alternatives.

Jay Pfaffman earned his Ph.D. at Vanderbilt University and is an assistant professor of instructional technology at the University of Tennessee. He teaches graduate courses in using technology in educational settings. His research interests focus on how best to use computers in schools and how to make instruction more engaging. He is the developer of Webliographer, an open source tool for sharing web resources.

References

Acohido, B. (2002, May 13). Microsoft pitches schools new licensing option. *USA Today*. Retrieved February 5, 2007, from http://www.usatoday.com/tech/news/2002/05/13/schools-microsoft.htm

Apple. (1987). *Apple human interface guidelines* (3rd ed.). Boston: Addison Wesley.

Baker, M. (2004). *The Mozilla project and mozilla.org*. Retrieved February 7, 2007 from http://www.mozilla.org/editorials/mozilla-overview.html

British Educational Communications and Technology Agency (BECTA). (2005). *Open source software in schools: A study of the spectrum of use and related ICT infrastructure costs*. Coventry, UK: BECTA. Retreved March 8, 2007, from http://ferl.becta.org.uk/display.cfm?resID=11479&page=633&catID=199

Cave, D. (2001, July 10). Microsoft to schools: Give us your lunch money! *Salon.com*. Retrieved February 5, 2007 from http://archive.salon.com/tech/feature/2001/07/10/microsoft_school/index.html

The GNU Project. (1996). *The free software definition*. Retrieved November 5, 2006, from http://www.gnu.org/philosophy/free-sw.html

Krebs, B. (2007). *Internet Explorer unsafe for 284 days in 2006*. Retrieved February 7, 2007 from http://blog.washingtonpost.com/securityfix/2007/01/internet_explorer_unsafe_for_2.html

Leadbeater, C., & Miller, P. (2004). *The Pro-Am revolution: How enthusiasts are changing our economy and society*. London: Demos.

Perkins, M., & Pfaffman, J. (2006). Using a course management system to improve classroom communication. *The Science Teacher, 73*(7), 33-37.

Pfaffman, J. (2005). Open source solutions: Moodle. *Learning and Leading with Technology, 33*(2), 42-45.

Raymond, E. S. (1997, Nov). The cathedral and the bazaar. Retrieved February 8, 2007, from http://www.catb.org/~esr/writings/cathedral-bazaar/

Stallman, R. (1985). *The GNU manifesto*. Retrieved November 13, 2006, from http://www.gnu.org/gnu/manifesto.html

Wikipedia. (2006). Retrieved November 6, 2006, from http://en.wikipedia.org/wiki/Wikipedia

Zetter, K. (2002, June 4). Schools cry bully over Microsoft licensing fees. *Computerworld*. Retrieved January 15, 2004, from http://www.computerworld.com/softwaretopics/os/windows/story/0,10801,71690,00.html?nlid=PM

Zymaris, C. (2003). *A comparison of the GPL and the Microsoft EULA*. Retrieved March 8, 2007, from http://www.cyber.com.au/about/comparing_the_gpl_to_eula.pdf