Free Software in Education: Is it a Viable Alternative?

Amatriain, Xavier
Griffiths, David
Universitat Pompeu Fabra
xamat@iua.upf.es
david.griffiths@upf.edu

Abstract

This paper reviews the advantages suitability of Free Software with particular regard to education. The characteristics of Free Software and Open Source and their respective licenses are explained. The pragmatic advantages of Free Software are outlined, with a focus on technical, economical and business related issues. It is proposed that the choice of software is not morally neutral, and that free software is a more coherent choice. The education specific aspects of Free Software are examined, and arguments put forward as to why Free Software is the best option for education and research. Practical guidance is provided on Free applications which can substitute for proprietary applications. Finally a four step route map is proposed for individuals or institutions who wish to move from a proprietary to a Free software environment.

1. Introduction

Free software is relevant to the issue of globalization in three principal ways. Firstly it provides an alternative to the economic model which has led to a very high degree of concentration of power over operating systems and applications in the hands of a very small number of large corporations. It is hard to establish reliable figures for installations, but it is often estimated that Microsoft Windows is installed on approximately 90% of the worlds computers, for example [1]. Secondly, both Free and Open Source software encourage localization into many languages. This can be done by volunteers who have access to the source code, whereas localization of proprietary software normally depends on a commercial decision by the developer. This exemplified by translate.org.za initiative in South Africa, which has made translated Open Office which has made the Mozilla browser available in Xhosa, Zulu, Venda, Sotho and Siswati. The Indian Linux Project is preparing a Linux distribution that supports Indian Languages at all levels [2]. It is also notable that OpenOffice, with similar functionality to Microsoft Office, is available in 42 languages on its download site, [3]. while Linux is available in 25 language distributions [4]. Finally, it provides an attractive cheaper alternative for countries and institutions which cannot afford to pay for updated commercial software which requires ever more powerful hardware to run it. This is true not only for the third world, but also for the countries of Eastern Europe, and for a number of sectors in the developed world. The education community in particular has shown interest in these practical advantages and other ethical and pedagogic aspects, for example the initiative in Extremadura in Spain [5]. This is the aspect of Free software which we principally focus in this paper, providing an introduction to the basic concepts, describing the `pragmatic and ethical advantages of Free Software in this context, and proposing a route map for moving towards a Free software environment.

1.1. The four freedoms

In this paper we use the term Free Software in the sense defined by the Free Software Foundation. According to this widely accepted definition, in order to qualify as Free Software any piece of software must observe the following four freedoms [6]:

- Freedom 1: Unlimited use for any purpose
- Freedom 2: Freedom to study and modify it for your own needs
- Freedom 3: Freedom to copy it
- Freedom 4: Freedom to distribute it with any modifications.

It is clear that in this sense “free” is not used as meaning “no economic cost” but rather as granting specified freedoms to its user. Because of this Free Software is often referred to as “Libre Software”, using the Spanish word “libre” with its less ambiguous meaning. The opposite of Free Software is Proprietary Software. It is important to stress that Free Software is not necessarily non-commercial, and that neither does the absence of payment for use mean that software is Free in the sense we have defined (for example Microsoft Internet Explorer).

Freedom in Free Software is established through copyright (which in this case is referred to as “copyleft”). Copyright allows its owner to license the product, and it is in this license that the four freedoms can be granted and guaranteed. It is therefore a myth that the Free Software movement is opposed to copyright and authorship, on the contrary, it is one of the main goals of the Free Software Foundation to protect software authors and their copyright.

Hundreds of licenses are available for publishing a piece of Free software, and there are many others which seem to be Free but are not, because they do not grant the 4 freedoms defined above. In case of doubt the Free Software Foundation provide guidance on whether they consider a particular license to be Free [7]

In a broad sense, Free software licenses can be classified into 3 categories: GPL-like, LGPL-like and BSD-like.

GPL, the Gnu Public License [8]provides the strongest protection both for the author’s decision to make the software Free, and for the freedoms granted. The basic characteristic of GPL-like licenses is that all derived software or applications that use the licensed software must use the same license, thus guaranteeing that all related software will remain Free. Thus GPL “vaccinates” software against non-Free usage (although GPL’s detractors choose to describe it as “viral”).

The LGPL or Lesser GNU Public License is similar to the GPL but does not protect the software from being used in a proprietary application. In this sense it provides less protection for freedom and the author’s rights.

Finally the BSD or Berkeley Source Distribution license [9] is the least protective as it allows the licensed code to be used or derived in any way, be it in Free or in proprietary software.

1.2. Open Source

The terms Free Software and Open Source are often misused and confused. Although “open source” per se refers only to the fact that the software is distributed with its original source code available, the term is currently related to the Open Source Initiative (OSI) that was started in 1998, (see [10] and [11] to see how both organizations compare their initiatives). The Open Source Initiative advocates abandoning the “ideological” issues informing Free Software, and concentrates on the practical factors that make open source software more reliable from a technical point of view.

1.3. Personalities

Three people have greatly influenced the discourse surrounding Free Software, and have served as a focus for the range of positions taken on the issues: Richard Stallman [12], Eric Raymond [13], and Linus Torvald [14].

Richard Stallman is the founder of GNU and the Free Software Foundation. In 1983 he initiated the GNU (acronym of GNU’s Not Unix) Project for developing a Free UNIX-like operating system. In 1985 he defined Free Software described above, and in that same year he introduced the GPL license.

Eric Raymond is the founder of the Open Source Initiative. He started the project in 1998 after disagreeing with Stallman’s “utopic” view on the subject.

Finally, Linus Torvald is probably the most well-known personality related to Free Software, and is the creator and main maintainer of the Linux kernel [4]. In 1991 the Linux kernel was incorporated to the developing GNU system, which still lacked a kernel, and gave place to the GNU/Linux operating system used by millions of people around the world [15]. The Linux kernel is licensed with the GPL license.

2. Pragmatic advantages of Free Software

Author In this section we review the main practical advantages of Free Software, leaving the more ethical or social issues for next section, in which we concentrate on Free Software in education. The practical advantages can be divided into three broad areas: technical, economical and business related.

First, we concentrate on the technical advantages, which can be summarized as better quality, more security, more stability, collective ownership and effort, and price.

Free Software can achieve higher quality because the code can be revised by many different programmers with a wide range of perspectives. As a result the adopted solution will probably be the best in the technical sense, not being subject to obscure strategic or business-related issues. Needless to say, releasing a Free software package by no means guarantees immediate collaboration from developers, but the opportunity is made explicitly, and there is a natural tendency for useful and well designed packages to attract the largest developer communities.

Free Software is also more secure. This may not seem obvious: how can software whose code is completely accessible be secure? The technical reasons are complex to explain but can be understood through the “key&lock” metaphor. Knowing all the technical details of a lock, and even being able to build one, does not mean that an existing lock can be opened.

The same is true of cryptographic algorithms and security policies. On the other hand, having many developers look into the code as potential users increases the likelihood of that any security holes will be swiftly identified.

Another widespread misconception regarding Free Software is that it is less stable than proprietary software. It is certainly true that many software packages evolve so fast that it is difficult to keep up to date, but it is entirely up to users to choose if they want to upgrade to the latest version or not. On the other hand a piece of Free Software is guaranteed never to disappear. It is very common for a business to buy proprietary software and then find out some years later that the original developing company has disappeared and no support is available. This never happen with Free Software, because if the original developers are no longer available, another software company can provide support after looking at the source code.

The final technical advantage is cost. While Free software does not necessarily have to be non-commercial, but in the In the Free Software model, users pay for services and support. There are many companies which run successful businesses on this basis, for example [16], [17] and [18]. This cost is always easier to evaluate than that of a closed product which has a license price corresponding to opaque strategic calculations.

These technical advantages apply equally to non-Free but Open Source software, but is often suggested that the Free Software model is not compatible with general economical interests and disastrous for technology companies. In the following paragraphs we will argue that this is not the case.

Firstly, let us examine what makes proprietary software so prone to favor monopolies. When using a given proprietary tool I implicitly require all the people who professionally relate to me and with whom I have to interchange data or knowledge to use that same tool. If those people adopt the tool, then a transmission chain rapidly spreads this requirement, and before long a whole economic sector or region has to use the same software tool. Frequently this particular tool only works well with other tools from the same company, and this intensifies the effect. Eventually the monopoly also extends to hardware platforms. Furthermore, in many cases the monopoly holders are foreign companies, so all the money companies and users invest in software leaves the country or region.

In contrast, Free Software is independent and promotes equal rights, and all companies are invited to take the source code, study and sell services related to it. As a result, Free Software promotes the creation of a local and maintainable software industry, making markets flexible and adaptable and favoring national economy.

Free Software does not mean the death of the software industry (see [19] or [20]). Even at present the percentage of software engineers whose salary directly depends on the sale of commercial proprietary software is small. Free software promotes the model of paying for services rather than licenses, and so builds on this existing revenue flow. This can be a commercial model, as is shown by the fact that major companies such as IBM are already working in this direction (ref: IBM strategy)

Although all the pragmatic reasons mentioned so far are of great importance, others may be of even more significance, and we will discuss them in the context of education and universities.

3. Ethical and social advantages of free software

Access to software is increasingly a determining factor in people’s ability to communicate, learn, and work. We access the internet through software, we read, write and study through software, and when we look for a job we are required to have software skills. Furthermore, as more and more activity is centered around software related services, software itself is bound to condition the vision of the world for our new generations. The “windows” paradigm [21], for instance, has already modified the way we understand information, the way we think and the way we relate with the world.

From this perspective, the key issue is who is taking the decisions which decide the way that coming generations will understand the world? It seems clear to us that such issues are too important to be in the hands of a single company with no social commitment other than maximizing shareholder value. Depending on ones political view of the world two different possible situations may be identified. In the first vital software such as operating systems, or internet browsers would be the result of standardization processes that would at least set the basis or guidelines that could then be developed by different companies on common grounds. In the second situation, Free Software would be enforced and no proprietary or closed solution could be commercialized. We believe that both situations would at the end be fairly equivalent although our experience of some standardization processes leads us to think that the second possibility (enforcing Free Software) may be more practical and agile, but would also encounter more resistance and legal issues.

4. Free Software and education

In many ways the values espoused by Free Software also inform education in democratic societies: freedom of thought and expression, equal opportunities, community effort and benefit, without which education becomes indoctrination. A key factor in learning is access to information, and the state intervenes in this respect by providing public schools and libraries in most countries. Today, not only are documents stored as software, but, more importantly, software provides the main access channel to information. Moreover, software increasingly provides the environment within which learning is carried out.

Democracy education institutions have a duty to present a range of points of view on any given issue, and to promote critical thought and discussion. In order to respect this plurality one should avoid promoting the idea that there is only one effective software solution, and that this solution belongs to a particular company. There are however more significant underlying issues. Firstly, the software infrastructure and tools have a significant impact on the organization of institutions and the processes which they support, and consequently they should be under the control of the communities which use them. Secondly, in universities we aspire to "teach to learn". Proprietary software discourages this because it withholds information from users, and does not allow them to explore the potential of the software or to extend it. It may appear that this argument exaggerates the importance of accessing the source code, as it may be suggested that only professional software engineers will be able to really make use of it. There are, however, many levels of interaction with software, as is shown by those web authors using proprietary tools who end up editing HTML. Free software can ensure that appropriate tools are available at all levels so that users can take control of the environment which they use.

The practical advantages described in the previous section also apply in an educational environment: a school or university can profit from lower costs, and provide more effective support for the national or local economy. An additional practical issue for schools is that important for learners to have the same software at school and at home. When proprietary software is used, most students are unwilling or unable to pay for the software and so they use unauthorized copies. In this way the education system becomes complicit in the use of illegal software.

Finally, many of the arguments in favor of Free software can also be applied to Free educational content, and a number of initiatives have been established which take this approach [22], [23].

4.1. Free Software and Research

Research is an important activity in many educational institutions, especially at the university level, and Free Software has a number of advantages in this respect.

The scientific method in itself depends to a great extent in the free exchange of knowledge and ideas. Scientists and researchers are realizing more and more that current duplication of effort because of fear of losing intellectual property makes no sense, and hampering research and scientific progress. In many cases, however, publishing research results or the algorithm is not enough, as today the implementation is often an important part of the result. We need to know whether a particular algorithm can work in different systems, how it performs in different situations, and its speed in different implementations. Software can be, and in many cases is, considered another sort of "publication".

With Free Software we are inviting others to compare, use and enhance our results. And more importantly, we are contributing to the preservation of human knowledge.

5. Free applications in an educational environment

In this section we briefly comment on some Free alternatives to tools which are widely used in education, in order to demonstrate that Free Software is a viable option. We refer readers interested in substituting a particular application to the Free software listings such as that maintained by the GNU project [24], the JOIN! project mentioned above, or to simply search the Internet for "Free Software Application X".

Probably the most widely used application in education, and beyond, is Microsoft Office. This proprietary package is used by staff, faculty and learners alike. A number of Free applications exist that substitute particular components in the package. For instance, Abiword [25] is a solid replacement for Microsoft Word; Gnumeric [26] can substitute Microsoft Excel quite well;. MagicPoint [27] is a replacement for Microsoft Powerpoint; and TotalRekall is a reasonable substitute for Microsoft Access, though rather complicated to install in its Free version. There is, however, a Free package which substitutes for the whole of Microsoft Office (except for Access, included only in Microsoft's Professional edition): OpenOffice [3]. This is a complete suite that can be used to edit text documents, work with spreadsheets or prepare overhead projections. The package is very robust, stable, and comprehensive and many new features are planned to be added in the short term. The user interface is so similar to that in Microsoft Office that the learning curve for an MS Office user is almost non-existent. Furthermore, OpenOffice has been translated to a many languages, including some not available for Microsoft Office. OpenOffice is a cross-platform application that can be installed in both GNU/Linux and Microsoft Windows.

Another set of tools of general use, but also important in education, are the various internet services including web browsers and e-mail clients. There are many Free tools in this area but Mozilla is of particular interest. Netscape gave this Free version to the community and has now announced that the proprietary version is soon to be discontinued. The Mozilla package includes a web browser, an e-mail client and a basic webpage editor. The lighter Firebird version includes only the web browser. The Mozilla web browser and e-mail client is more secure and technologically advanced than Microsoft's Internet Explorer or Outlook Express.

In scientific or technological education it is very common to use mathematical packages such as Matlab. Octave [28] is a good Free substitute which is used in many research institutions, and lacks only some Matlab toolboxes which are not included in the regular version.

For graphics a number of applications can be recommended. The Gimp [29] is a graphics and photograph editor which is very similar to Photoshop, and is even more advanced in some specific functionalities. For drawing diagrams and vector graphics Dia [30] and Xfig [31] can be used as substitutes of Microsoft Visio.

Many other Free tools can be used in education including some very advanced applications published as Free software by universities, or educational/recreational software for small children, such as TuxPaint [32].

Two examples of applications which are only available as Free Software are LaTeX and Emacs. The former is a complete environment for generating professional quality texts for publishing, and it includes tools for handling bibliographies or a large collection of publishing styles. LaTeX is even required by some international publishing companies when handing in manuscripts. Emacs is a very well-known tool, developed and maintained by Richard Stallman. It is a very advanced text editor that has tools for editing different formats such as LaTeX or html.

All these tools can be installed in both GNU/Linux and Microsoft Windows. Most have native Windows versions that can be installed just like regular Windows applications, while the others can be run using the Cygwin environment [33], which runs a GNU/Linux simulation under Microsoft Windows.

Finally, a number of Free Software e-learning environments have now been developed, and 48 are listed in [34].

5.1. A Word on formats

One of the fundamental problems related to the use of proprietary software is that of file formats, which is probably one of the main factors in the emergence software monopolies. Although most Free applications are now capable of opening proprietary formats (following substantial development effort), nevertheless, when we send someone a proprietary document (e.g. a Microsoft Word file), we are telling them to use the same application (ref: No word attachments). Additionally, because the file format is not known, we cannot be sure of what information is encoded in it. The file could, for example, be attaching personal information without our consent. The problem is that in many important application areas, such as editing formatted text, there are no standard formats, and this is a need which international standardization organizations such as ISO could address with advantage.

What formats, then, can be used in place of a Microsoft Word file? A first alternative is an ASCI text file, and although in a text file all formatting is lost, this is the best format for many applications. Many people are used to attaching Microsoft Word files for simple texts that could well be included in the body of the mail or as a separate text attachment, which is also much smaller.

If on the other hand, it is important to keep the format for a text file, one has to decide if it is to be editable. If this is not necessary then a Free format such as postscript or PDF may be used (although the latest versions of PDF are not Free). If the text file is to be editable a good option is html, which is Free and highly efficient. It is, of course, also possible to use a Free format related to an application such as Open Office. In this case it is true that we force receivers to use a particular application that they may not have, but this is clearly better than using a proprietary application, as the receiver can download the Free application.

Finally if what a Free editable professional format for publishing is required, LaTeX is the best choice. This should have been the case, for instance, in the template of this conference, and offering only a Microsoft Word format has forced this author install the software specifically to write this paper.

Another area in which formats are important is the Internet. We have already mentioned that there are many different Free web browsers, but the truth is that it is not always easy to browse the internet with some of them. The main reason is that many web pages do not using standard html code, but rather code extended with some proprietary additions. This is particularly worrying as Internet is supposed to promote free access to information and now it turns out that this information can only be accessed for an application belonging to a particular company. The case of Macromedia Flash is also a direct attack on free access to information ([35]) although it has to be admitted that there is still no clear Free alternative to it. When browsing the Internet we must demand the right to access any web page through any browser, and punish companies that force the use of a particular one (usually Microsoft Internet Explorer) by not using their services.

6. A proposed route map to the adoption of Free Software

When deciding whether to adopt Free Software for an institution it should be remembered that this is not an "all or nothing" decision. Free Software can be approached carefully in order to lessen the fear factor. We now propose a route map to go from an "all proprietary" to an "all Free" situation in small steps that will minimize the sense of sudden novelty and insecurity. This guidelines are intended to help both institutions and individuals that are willing to support and use Free Software, but are afraid of doing so through a sudden "revolution" and would rather think of it as a "transition".

Before describing the route map, we need to respond to users who ask "Will I be able to do everything I can do in my "all proprietary" environment when I reach the "all Free" environment. The short answer is "yes,". The long answer is there is a Free alternative for almost 100 percent of proprietary software (excluding games), including of course common applications such as office suites or internet browsers. In general terms, the state and features of these alternatives is comparable to the proprietary applications. Nevertheless, one should recognize that in some instances certain features and functionality that will have to be given up. Should this be the case then this must be balanced against the advantages which we have described above, and the availability of Free Software of some particular applications which are not available in proprietary environments.

We now outline the proposed route for an average user who currently works with Microsoft Windows and uses no Free Software.

6.1. First contact with Free Software

The first step we propose is to install Free alternatives to those applications we use the most, so as to provide an initial contact with Free Software. The same operating system can be used, and there is no need to get rid of any current applications, just to add the new ones and progressively start using them. This does not require much effort from the user.

It is probable that the most used applications are Microsoft Office and the Internet browser, and if so a good start would be to install the Windows version of both OpenOffice and Mozilla.

6.2 All applications with a Free alternative

By continuing with the process of substitution, users should reach the point where most, if not all, the applications they use have a Free alternative in the system. An effort is required to use the Free versions whenever possible so as to become used to them.

6.3. Installing GNU/Linux

The next step is to decide on which GNU/Linux distribution best fits the users needs (see previous section) and install it on a separate partition or hard-drive in the system. In this way, using Lilo or similar software, a dual booting machine can be configured, offering the user a choice of booting in GNU/Linux or Windows. A good option at this point is to have a third disk partition where all data that should be accessible from any of the two operating systems is kept.

The newly installed GNU/Linux partition should have all the various Free applications that were previously installed on Microsoft Windows, so as to minimize the sensation of novelty and keep work practices stable. Again, the user should make an effort to launch GNU/Linux whenever possible.

6.4. An all-Free environment

Having reached the situation described in 6.3, it is a matter of time before the user is ready to switch to an all-free situation. Free Software under GNU/Linux can then be used nearly all the time, leaving the Microsoft Windows partition only for very exceptional cases such as writing a paper for a conference that only provides Microsoft templates.

7. Conclusions

In this paper we have outlined the main advantages of Free Software with particular reference to the educational environment. As people who are responsible of some part for this academic community we have the duty to defend the independence of education, and promote freedom of thought and acquisition of knowledge. As we have shown in this paper, Free Software promotes all this and more.

Various parts of society are already starting to realize the advantages of Free Software and are moving toward it. On this important issue educational institutions and universities should not be in the last wagon, but should rather take a stand, and adopt a leading role in this transformation.

Free Software requires a minimal personal investment and in return it provides many satisfactions and advantages. Our choice of software is no ethically neutral, and Free Software can reassure us that we have chosen a coherent and morally defensible solution.

10. References

[1] Wikipedia, Microsoft Windows.
[2] Indian Linux Project.
[3] OpenOffice.org, Open Office Community Conference homepage.
[4] Linux, Linux Online.
[5] Junta de Extremadura, Linex homepage. 2002.
[6] Free Software Foundation (FSF), The Free Software Definition. 1996-2004.
[7] Free Software Foundation (FSF), Licenses. 1996-2004.
[8] Free Software Foundation (FSF), GNU General Public License.
[9] Open Source Initiative, Berkeley Source Distribution License., Berkeley University.
[10] Free Software Foundation (FSF), Why "Free Software" is better than "Open Source".
[11] The Open Source Initiative (OSI), How is Open Source related to Free Software? Frequently Asked Questions. 2004.
[12] R Stallman, Interview - Free Software, Free Society!. 2004, Scotland independent media centre.
[13] Eric S Raymond, The Magic Cauldron. 2000.
[14] L Torvald, Linux commercial use. 1993, Cited in History of Linux, Short, Chris; Short, Carri.
[15] Harald Alvestrand, Estimating the number of Linux users., The Linux Counter.
[16] MySQL, MySQL homepage. 1995-2004.
[17] Trolltech, Trolltech homepage..
[18] Redhat, Redhat homepage.. 2004.
[19] The Open Source Initiative (OSI), Open Source Case for Business. 2004.
[20] K Kenwood, A Business Case Study of Open Source Software. 2001, The Mitre Corporation.
[21] A. Kay, Lecture at Georgia Tech, Atlanta "The Computer Revolution Hasn't Happened Yet". 1997, Radio IDT.
[22] Massachusetts Institute of Technology,
[What is MIT OpenCourseWare? MIT OpenCourseWare FAQs.. 2003.
[23] M Cedergren, Open Content and Value Creation. First Monday, 2003. 8(8).
[24] Free Software Foundation (FSF), FSF/UNESCO Free Software Directory. 2002 - 2004.
[25] AbiWord, Word Processing for Everyone. 1998-2004.
[26] GNOME Foundation, Gnumeric. 1999-2003.
[27] MagicPoint, Official homepage.
[28] Octave, Homepage. 1998.
[29] The GIMP Team, The GIMP Homepage. 2001-2004.
[30] Dia, Homepage.
[31] Xfig, Homepage. 2004.
[32] New Breed Software, Tux Paint. 2002-2004.
[33] Cygwin, Homepage. 2000-2003.
[34] S Leslie, OpenSourceCourseManagementSystems. 2004, EdTechPost.
35] J van Baal, Macromedia Flash considered harmful. 2004.

Acknowledgement

The authors wish to thank Georg Greve, founder of the Free Software Foundation Europe (www.fsfeurope.org), for the ideas and materials which he supplied for this paper.