Promoting Open File Formats and Open Source at Dartmouth

Presented to the Council on Computing

May 29, 2003

Summary

The College is called upon to promote the use of open file formats for all documents which will be publicly archived or exchanged within the Dartmouth Community. The College is also called upon to provide support for and promote the use of Open Source applications as a means to achieve this goal. Finally, the College is called upon to begin to provide support to those individuals who wish to use an Open Source operating system such as Linux.

Support and promotion of open file formats (especially when achieved through the use of Open Source applications) will have a positive effect on finances, licensing agreements, long-term preservation of public documents, and represents a posture consonant with the endeavors of an academic institution. The changes required by adopting this position are relatively small in scale, and more than compensated by the positive effect engendered.

1 Introduction

This presentation to the Council on Computing was prompted in part by the publication of series of resolutions [1] which were passed by the University of Buffalo faculty on April 1, 2003. Those resolutions (in order of appearance) were for institutional support for the GNU/Linux operating system, Open Source software, and open document formats and communication protocols. In this presentation to the Council on Computing, we raise basically the same issues, but with a slightly different emphasis.

We wish to present and discuss the following five points:

- What do we mean by open file formats and Open Source software?
- Why is the use of open file formats and Open Source software better in general, and in particular for Dartmouth?

- What are the costs to Dartmouth of supporting these Open Source initiatives?
- What are the costs to Dartmouth of not supporting these Open Source initiatives?
- What are some specific pieces of software which utilize and produce open file formats, can easily be introduced into the student/faculty population, and which should be easy to support?

2 The Discussion

2.1 What do we mean by open file formats and Open Source software?

An excellent starting place for information about open file formats is available locally on `linux.dartmouth.edu` [2], but as a working definition, we take an open file format to be a file (document) type whose specifications are public and which is unencumbered by patent restrictions. TEXT, HTML, XML, and PDF are good examples of open file formats; a Word document or Adobe Photoshop document are examples of documents based upon closed or proprietary formats. Anyone who wishes to open a Photoshop document must own a (current) copy of Photoshop, however to read an HTML document, any browser in the world will do.

To see some of the implications of this distinction, we quote from Adobe's web site [3]: Since PDF has an "open file format specification, PDF is available to anyone who wants to develop tools to create, view, or manipulate PDF documents. Indeed, more than 1,800 vendors offer PDF-based solutions, ensuring that organizations that adopt the PDF standard have a variety of tools to leverage the Portable Document Format and to customize document processes."

In contrast, the number of vendors who can read the format specification for a Word document is one, and it is a remarkable effort of reverse-engineering which has allowed Open Source applications like OpenOffice.org to be able to import and export virtually any Microsoft Office document.

The official [4] definition of Open Source software has ten points, which we make no attempt to cover even cursorily here. Very loosely speaking, Open Source software is not simply software to which the source code is available, but also carries with it rights for redistribution of the software in its original and modified forms.

Closed source or proprietary software does not reveal the source code, and generally restricts the rights of its users, both in terms of redistribution and reverse engineering, and also in terms of (multiple) installations.

As a simplistic analogy, imagine a car sold to you with the hood welded shut. You put the key in the ignition, turn it, and generally the car runs. You haven't a clue how the car runs, but it does. Now after a while, it doesn't seem to run as well as it did before. What

can you do? Not much; you've got the closed source model with the welded hood. With the Open Source model, not only can you open the hood, but attached to the fender is an instruction manual (and perhaps a list of mechanics) for how to fix the car. Finally, you grow tired of the car, but under the terms of your closed source car purchase, you can't even give it away, so it sits rusting in your backyard. With the Open Source model, you can rebuild the engine, tune the suspension, and pass it on to your sixteen-year old daughter.

2.2 Why is the use of open file formats and Open Source software better in general, and in particular for Dartmouth?

Points to discuss include:

- The use of open file formats is critical to any institution or person interested in preserving materials which will remain publicly readable for a extended period of time.
- Open Source represents a philosophy which is consonant with the academic endeavors of an educational/research institution.
- Open Source software is typically free, upgrades are free, and the software can be installed on as many machines as you like without violating any licensing agreements. Closed source products force expenditures not only for the creators of documents, but also for those who wish to read them.
- Open source software minimizes security risks for users, while closed source software more easily permits an invasion of privacy.

Among other things, the College is an information and knowledge distribution center. The Library is certainly the largest purveyor of such information, but the collective web content present at Dartmouth represents an ever growing contributor. Whether the information to which we offer access is current or of an historic nature, the intent is for these materials to be publicly available for an extended period of time. Virtually by definition, such materials should be "freely" available to the public, and that means at little or no cost.

The use of open file formats is critical to any institution or person interested in accomplishing this task, while the use of proprietary or closed formats hobbles this effort at several levels. By using a closed file format, not only does the creator of a document incur the expense of the software, but they demand of anyone who wants to read the document that they too buy that software. And of course their software version must be at least as new as yours, or they have little chance of reading the document. Finally, let us not forget that vendors of proprietary software (even those who remain in business) may not continue to support their products. When did you last open a MacWrite or MacPaint document, or a Hypercard stack? For that matter, ever try to read a Microsoft Word document with Microsoft Works? This forced incompatibility resists all attempts at making documents public in the true sense of the word.

With open file formats, many vendors can produce products which are capable of accessing materials, upgrades are freely available, and the chances of a public resource losing its usefulness is greatly diminished.

Open file formats and Open Source represent a philosophy which is consonant with academic endeavors. Faculty's goal as researchers is to make public the results of their research. Keeping research private may be necessary for national security, but outside that arena, a closed source philosophy generally forces people to reinvent the wheel time and time again. With closed source, one may like a feature of a piece of software, but are forced to invent a new scheme for accomplishing a task which has already been successfully completed. Surely a better use of time can be found. In contrast, Open Source software usually has a large user base, and there are many people actively working on aspects of its development. Since the source and development is open, those who provide computing services rarely reinvent the wheel when doing software development.

Open Source software is usually released under the GPL (GNU Public License)[5], which among other things preserves the user's rights to modify and redistribute the software. This allows a faculty member to obtain one copy of some piece of software and distribute it to an entire class. This is a particular advantage when the software is specialized, and may get used only once in a student's career. It allows all of us to install software on multiple personal machines with no violation of licensing agreement. It allows us to find a variety of applications which serve our general needs, and if necessary tailor them to fit our specific requirements. As a good paradigm of the use of Open Source software, consider a recent endeavor by the Math Department. They were interested in using software to track requests for computer assistance within the department. They were told they could use the same software as Computing Services (Remedy?), but the licensing costs were very high, and their needs modest. To be honest, they weren't sure whether such software would be an asset or a nuisance. They found several open source options, examined their capabilities, downloaded the most appropriate one, made minor modifications to the code to interface with their local security protocols, and installed it. All this in two days time. As it happens, this software required both PHP and MySQL, but these are both Open Source, freely available (and were already running on their servers), so this was no problem. Developing this kind of software from scratch would be clearly cost ineffective, and buying software which may have many more features than were needed, at an exorbitant cost, would have been foolish. The Math Department obtained a polished piece of software for essentially no investment. Moreover, ideas for improving the software have been bandied about, and may eventually be forwarded to the upstream authors.

Closed source software presents many other problems other than potentially denying public access to archived materials. It generally allows the possibility for an invasion of privacy. As a recent example, consider the "critical" software upgrade for Windows XP which required an update to their media player. From [6],

"This software upgrade contained a new EULA which grants Microsoft the unrestricted right to automatically alter your copy of Windows so that it will '... disable your ability to copy and/or play secure content and use other software on your computer.' "

From [8] (see also [7]),

In particular, the privacy problems with WMP version 8 are:

• Each time a new DVD movie is played on a computer, the WMP software contacts a Microsoft Web server to get title and chapter information for the DVD. When this contact is made, the Microsoft Web server is given an electronic fingerprint which identifies the DVD movie being watched and a cookie which uniquely identifies a particular WMP player. With this two pieces of information Microsoft can track what DVD movies are being watched on a particular computer.

• The WMP software also builds a small database on the computer hard drive of all DVD movies that have been watched on the computer.

• As of Feb. 14, 2002, the Microsoft privacy policy for WMP version 8 does not disclose that the fact that WMP "phones home" to get DVD title information, what kind of tracking Microsoft does of which movies consumers are watching, and how cookies are used by the WMP software and the Microsoft servers.

• There does not appear to be any option in WMP to stop it from phoning home when a DVD movie is viewed. In addition, there does not appear any easy method of clearing out the DVD movie database on the local hard drive.

This is certainly not an isolated example. The point is that when using closed source software, you have no idea what information is extracted about you or your system. With Open Source software, this threat is diminished because the code is public. E-commerce would collapse without Open Source. The US recently adopted the AES (Advanced Encryption Standard) which is a public standard and which is implemented by an encryption scheme called Rijndael, the details of which are public. The commerce department recently certified this method as the default method for electronic commerce. Can you imagine commercial banks using an encryption scheme the details of which were not public? How could they be sure the program was doing what the vendor said? How could they be sure there was no secret way of breaking the encryption? Only by making the code public, can these fears be allayed.

In summary open file formats and Open Source software are necessary for best ensuring long-term accessibility to public documents, represents a concept of open research mirroring that of an academic institution, allows the use and redistribution of software with a license that encourages the use and further development of that software, and which minimizes security risks and exposure to invasion of privacy.

2.3 What are the costs to Dartmouth of supporting these Open Source initiatives?

The impact of implementing the suggestions in this proposal is intended to be minimal.

- If the College chooses to support and promote the use of open file formats, then Computing Services needs to support applications which produce documents using these

formats. As probably the single most important example, consider OpenOffice.org as a replacement to Microsoft Office. OpenOffice is an Open Source application, utilizing the open file format XML as its underlying document format. It is a cross-platform application, able to run on Windows, Mac OS X, and various UNIXes. It is feature-comparable to Microsoft Office, and in fact can import and export the vast majority of documents produced by MS Office, a fact which may aid in transition from our current state. Being feature-comparable, all that consultants for Computing Services need to do is become familiar with minor changes in menus and features. They still will be answering questions about an office suite, just not Microsoft's. Indeed not too many years ago, the College converted wholesale to Corel's Office suite, and recently all administrators were converted to using Windows, so this kind of change is well within the realm of standard institutional change.

- Part of this proposal is to begin to provide support to those individuals who wish to use an Open Source operating system such as Linux. While it would be nice to have some consultants who could answer basic questions about using Linux on the Dartmouth campus, it is certainly not the expectation that all Linux distributions be supported. In fact, what is most lacking for basic Linux support is access to documentation for those looking for it. Most people who use Linux or experiment with it have very basic needs and questions: how do I set up a printer?, how do I access BlitzMail? Can I use instant messaging?, and so on. Quite frankly, documentation for how to do all these tasks exists, but is scattered about campus on various servers. Where it belongs is `www.dartmouth.edu/help/pdf`. Thus, preliminary computing support for Linux simply requires granting access to post documentation to that site.

- Another important cost is the need to establish a committee whose job it is to constantly review the features of "core" software used on campus. While OpenOffice.org is a very large software suite, perhaps smaller specialized alternatives would be preferable or included along with OpenOffice as part of Core Dartmouth software. For example, for a word processor, one might consider Abiword which is available for Windows, Linux, BeOS, and QNX.

  It is very important to remember, that because open file formats are used, a given application does not have to function on all platforms. It is only necessary that there exist applications which run on each platform which do utilize and produce documents implementing that file format.

2.4 What are the costs to Dartmouth of not supporting these Open Source initiatives?

The following is the executive summary of an article [9] titled "Your Open Source Plan":

> The open-source movement is helping turn significant chunks of the IT infrastructure into commodities by offering free alternatives to proprietary software. The promise of the past several years has begun to materialize as one by one

the hurdles to open-source adoption have dropped away. Major enterprises are running mission-critical functions on open-source IT. Big vendors have lined up to support it or port their applications to it. CIOs who have implemented it report significant reductions in total cost of ownership. Our conclusion? CIOs who don't come to terms with this revolution in 2003 will be paying too much for IT in 2004. To avoid getting stung, CIOs should pursue as least some components of this 2003 open-source agenda: Get your feet wet with relatively low-risk Internet applications. Investigate the new support offerings from Dell, IBM, Sun and others. Start replacing proprietary Unix hardware with less costly Intel systems running Linux. Standardize infrastructure, including Web servers and desktop systems. The more daring will move enterprise level apps like SAP to Linux platforms.

More specific cost issues concern:

- Continued licensing costs (e.g. Microsoft Campus Agreement).
- Public archives no longer readable by the public after a while, due either to formats no longer supported or to documents produced using a version of proprietary software newer than what Jane Q. Public has.

  Upgrades for the public are a real issue from which we in the College are somewhat insulated. Even within the Dartmouth community, it would be anarchy if the Dean's office upgraded their software, since two-thirds of the faculty wouldn't be able to read the distributed documents. It is of course not that Open Source software doesn't upgrade, or that upgrading software is any easier (if cheaper), but ensuring backward compatibility is a significant goal of Open Source software. What are the odds that Microsoft Office 2000 can handle all of MS Office XP's documents?
- Perhaps a continued or increased level of viruses spreading throughout the campus. Viruses are constructed to attack vulnerabilities in software. Open Source software has the virtue that many more eyes are examining code, and they have a far greater potential of pointing out vulnerabilities. And with more and more data formats having executable content designed into them (e.g. recent problems with Visual Basic), the problems inherent in closed source software will only grow, and our costs to get rid of viruses on campus will only increase.
- Wasted bandwidth. A document consisting solely of the words "Hello World!" is 12 bytes long in text, 4748 bytes long in OpenOffice (Mac OS X), and 19,742 bytes long in Word (Mac OS X). Do we really need to send 19,742 bytes to say hi? And if this is sent as mail, it takes up much wasted space on the mail server. From there (since most of us are a bit lax about cleaning out the inbox), the same file gets backed up to another disk or tape depleting yet another resource, for no particularly good reason.
- The lack of support of Linux (even to the extent of making documentation available which has been generated by others on campus), causes a large number of headaches,

especially for network services. All too often, I have seen notes from the network administrators complaining about misconfigured Linux mailers sending errant messages to root@dartmouth. Root@dartmouth is a real person who doesn't like his mailbox inundated because of ignorance in setting up a mail system. A little documentation can go a long way.

2.5 What are some specific pieces of software which utilize and produce open file formats, can easily be introduced into the student/faculty population, and which should be easy to support?

As general reference points, we have:

The OpenCD [10] "is a collection of high-quality Open Source Software. All of the programs on the disc run under Windows; the disc is intended to be an Open Source showcase".

For Linux users, "The table of equivalents / replacements / analogs of Windows software in Linux" [11]

For a few specifics we have

- OpenOffice.org [12] instead of Microsoft Office
- Mozilla [13] instead of Internet Explorer (N.B. Internet Explorer for the Macintosh has a well-documented problem in handling user-generated security certificates for secure web sites)
- OpenSSH instead of commercial implementations of SSH
  (Putty[14]/WinSFTP[15] for Windows, MacSSH/MacSFTP[16] for the Mac)

  Note: MacSSH is released under the GPL, but MacSFTP is only based on OpenSSH; it is a commercial product, but cheap.
- OpenLDAP and kerberos
- Mozilla/OpenOffice instead of Frontpage

Finally, it should be pointed out that software which runs only on Linux can still be used in a graphical environment on both Windows and Macintosh platforms using the Open Source software VNC (Virtual Network Computing) [17], developed at AT&T Labs in the UK. This software is used extensively in the math department, and is currently our main solution to accessing IP restricted web services via a home DSL or satellite connection.

References

[1] Faculty Senate, University of Buffalo, April 2003. http://orange.math.buffalo.edu/csc/resolution2_february2003.pdf.

[2] The Open File Format Project, http://linux.dartmouth.edu/off

[3] What is Adobe PDF?, http://www.adobe.com/products/acrobat/adobepdf.html

[4] Open Source Initiative, http://www.opensource.org/docs/definition.php

[5] GNU Public License, http://www.gnu.org/copyleft/gpl.html

[6] License to plunder,12 July 2002, http://www.infoworld.com/article/02/07/12/020715opestrat_1.html

[7] Microsoft Media Player logs users' DVD picks, 21 February 2002. http://www.itworld.com/AppDev/1471/IDG020221mediaplayer/page_1.html

[8] Serious privacy problems in Windows Media Player for Windows XP, 20 February 2002, http://www.computerbytesman.com/privacy/wmp8dvd.htm

[9] Executive Summary, Your Open Source Plan, 15 March 2003, http://www.cio.com/archive/031503/opensource.html

[10] The Open CD, http://theopencd.org/

[11] The table of equivalents / replacements / analogs of Windows software in Linux, linuxshop.ru/linuxbegin/win-lin-soft-en

[12] OpenOffice.org project, http://www.openoffice.org/

[13] Mozilla.org,http://www.mozilla.org/

[14] Putty, A Free Win32 Telnet/SSH Client,http://www.chiark.greenend.org.uk/~sgtatham/putty/

[15] WinSCP, a freeware SCP client, A Free Win32 Telnet/SSH Client

[16] MacSSH and MacSFTP, http://pro.wanadoo.fr/chombier/index.html

[17] Virtual Network Computing, AT&T Labs, UK, http://www.uk.research.att.com/vnc/