Promoting Open Source at Dartmouth

Presented to the Council on Computing

April 8, 2004

1 Abstract

Open Source represents a philosophy which is consonant with the endeavors of an academic institution. It affects (or at least should) the manner in which we conduct our research, the manner in which we teach our classes, and the manner in which we disseminate information to the world. Institutionally, we acknowledge its importance in a few discrete areas, but seem generally unaware of its true value and power, its ubiquitous presence, and the degree to which we at Dartmouth are woefully behind in incorporating it into the fabric of our institutional mission.

To adopt an Open Source perspective means addressing complex and potentially conflicting issues of intellectual property, needs and interests of faculty and students, and careful consideration of costs. To support such an initiative, we need an institutional mission statement which endorses this effort, and provides authorization for a phased implementation plan.

In this document, we focus attention on the importance of Open Source, outline many clear benefits of adopting an Open Source philosophy, and indicate many nationally recognized Open Source projects and their sponsors. With an institutional mandate to move forward with this initiative, we can become a significant contributor to — and a beneficiary of — this important effort.

2 Some notable Open Source Projects

The following is a list of a few Open Source Projects of particular note. This list is only intended to indicate some degree of the breadth and complexity of projects which are under development; it is certainly not meant in any sense to approximate a comprehensive list. In fact, the following recent statistic gives some sense of the scope of the Open Source movement.

SourceForge.net is the world's largest Open Source software development website. From their newsletter (31 March 2004)[1]

Alexa, a website now owned and operated by Amazon, keeps track of the traffic generated by the top 100,000 websites in the world. According to Alexa, SourceForge.net is now ranked #258 (out of 100,000) in the world in terms of sheer traffic. To put this fact in some perspective, during the past 12 months, SourceForge.net has surpassed IBM.com, Sun.com, and Cnet.com in daily pageviews.

- OpenCourseWare (`ocw.mit.edu`), supported by the Hewlett and Mellon Foundations.

  To paraphrase their blurb, they have materials from 700 courses online spanning 33 academic disciplines and all five of MIT's schools. The course materials contain syllabi, course notes, homework, lectures, video, and so on. Dartmouth's efforts with Blackboard pale by comparison, and the reasons are important, but will be discussed a bit later.

- The Sakai Project (`http://www.sakaiproject.org/about.html`) Some excerpts from their site:

  The University of Michigan, Indiana University, MIT, Stanford, and the uPortal consortium are joining forces to integrate and synchronize their considerable educational software into a pre-integrated collection of open source tools.

  The products of this project will include an Enterprise Services-based Portal, a complete Course Management System with sophisticated assessment tools, a Research Support Collaboration System, a Workflow Engine, and a Tool Portability Profile as a clear standard for writing future tools that can extend this core set of educational applications.

  (my emphasis) *This effort will demonstrate the compelling economics of software code mobility for higher education, and it will provide a clear roadmap for others to become part of an open source community.*

- OpenOffice (`www.openoffice.org`) A complete Office suite which we discussed last year.

- OpenScience (`www.openscience.org`), funded by Alfred P. Sloan Foundation

  Open Source software in Acoustics, Forensics, Anthropology and Archaeology, Artificial Life, Astronomy, Aviation and Aeronautics, Chemistry, Cognitive Science, Complex Systems, Computer Science, Earth Sciences, Energy, Engineering, Geography, Information Technology, Life Sciences, Mathematics, Measurements and Units, Nanotechnology, Physics, Required Reading, Space, Tools.

- Open Source Systems for Libraries (`http://www.oss4lib.org/`)

  General Open Source Library Tools, plus a discussion of the conversion of the library system at Arizona State University-West (branch campus with 7000 students) to Linux-based system with the following highlights and technical specifications [3]:

  Highlights

  - Diskless workstations

- Linux OS for clients and servers
- Build vs. Buy for clients and servers
- Original software authoring
- Reusable documentation
- Scaling environment from workstation-class to enterprise-class
- Redesign to include open-source and open-architecture models
- Provide positive ROI and lower TCO
- Position services for future expansion and integration

The library workstations offer ASU and non-ASU users access to Internet resources using the Mozilla browser, access to floppy, Zip 250, and network storage, two virtual desktops, Microsoft file viewer applications through the use of WINE, and a disk management tool. The workstations take advantage of network PXE booting and receive a clean disk image into RAM. The workstations utilize Kerberos authentication and LDAP directory services for accounts. Since the disk image is in memory, once the power is dropped to a workstation, all footprints are erased. When the workstation starts up, a clean image is served to the workstation.

- Open Source Applications Foundation (`http://www.osafoundation.org/`)

  Main Project: Chandler, An Open Source Personal Information system like Evolution for Linux

3 Thorny Issues

3.1 Intellectual Property

One of the issues most frequently raised in opposing Open Source initiatives centers around the issue of intellectual property. This is clearly worth examining as it currently manifests itself at Dartmouth.

As an example lets consider some differences between MIT's OpenCourseWare site and Dartmouth's Blackboard site. As mentioned above, MIT's site contains materials from 700 courses online spanning 33 academic disciplines and all five of MIT's schools. The course materials contain syllabi, course notes, homework, lectures, video, and so on. This is a site completely open to the world.

While I have not checked the statistics, I am sure Dartmouth's site does not have the volume of entries that MIT's does, but that is not the main issue. Many of our sites have highly restricted access. In many of the course pages on Dartmouth's Blackboard site, guest access will get you nothing. Why? Well, good question. Many faculty put forth issues of intellectual property or copyright restrictions regarding republication of material. Perhaps

these are valid, but MIT manages to put a vast array of materials online. If they can do it, why can't we?

Regarding issues of intellectual property, I have heard faculty quoted as saying, "I worked for years on this syllabus and I'm not going to give it away". Well, (IMHO) if it's such a great syllabus, get someone to publish it. The whole point of an Open Source perspective is that generally you get back a great deal for what you put in. No, it is not a pyramid scheme, but give that syllabus to a hundred people and in three years you will see variations and connections you never dreamed of. For faculty to believe they a lock on this kind of creativity is absurd. But this kind of attitude adjustment from the faculty has to come down as a mission statement from the institution, not from the Council on Computing.

While not strictly speaking an open source project, a personal anecdote may be of value. Occasionally, I get asked by graduate students to describe an area of number theory known as class field theory. It is an important area, but one that is fairly advanced so gets taught infrequently. But to summarize it succinctly is quite difficult. In response to one person's request, I decided to give a series of seminar talks on the subject, and wrote up my notes for their reference. As usual when I have made the effort to produce some polished notes, I put it up on my web site for other graduate students to access in the future. A couple of years later, I was preparing to teach a course in this area, so I "googled" for class field theory, and noticed my name popping up in numerous links. I checked some of these pages and discovered various people had collected numerous survey articles on the subject with differing perspectives and various levels of detail. What a tremendous resource for me to find.

The point is that no matter what the endeavor, putting your ideas in the public eye almost always returns new ideas and perspectives to you. In software design, it is the difference between having a bunch of developers sit around trying to imagine new features you might want in an application versus having the consumers tell you what they need and want. The later generally produces a more useful product. For example, who among us would have thought the bouncing paper clip or bowing wizard was an essential feature of an office suite?

3.2 Software and verifiability of experiments in Science

Much experimental science is based on simulation, but simulations are typically produced by computer programs. When a scientist announces a new result, the first rule is that of reproducibility. I assume we all remember "cold fusion" a number of years ago.

The following are some excerpts from a 1999 (5 years ago!) article in the Chronicle of Higher Education [2]:

"Good science must be verifiable", said J. Daniel Gezelter, an assistant professor of chemistry at the University of Notre Dame [and founder of the openscience.org website]. But scientific research that relies on software to analyze or simulate scientific processes, he said, "is not verifiable in practice unless you can look at the source code".

But in fact, the use of Open Source software alone may not be sufficient to ensure verifiability. From [4], we quote:

> But the use of open-source software is insufficient. If the future of science depends on scientists' use of open-source software, one can very well argue that colleges and universities are under a positive obligation to move away from closed-source computing infrastructures as well as closed- source software. Consider this: many of the instructions in computer programs do little more than issue directives to the operating system; this is done by means of the operating system's application programming interface (API). To verify scientific software fully, the scientific community may need to examine the program's interaction with the operating system. Yet Microsoft refuses to document the Windows API fully and regards the Windows source code as an immensely valuable trade secret. What is more, Microsoft has taken the lead in lobbying for proposed changes to the U.S. commercial code that would effectively criminalize reverse engineering.
>
> It's not enough for scientists to use open-source software; they must also use an open-source operating system. Colleges and universities can help to assure the ubiquity of open-source software and operating system usage in science by moving to Linux as an international standard for academic computing.

However, many people do not want to release source code for various reasons, some of which probably include a fear that knowledge of their algorithm may allow others to leap-frog ahead of them. Perhaps, but we have tried to make the case that you generally gain more from an Open Source perspective than you relinquish. More often than not, others in academics are not out to leap-frog over you, but would welcome collaboration with you, and that serves both parties equally well.

A relevant quote [2] from a former Dartmouth Mathematics instructor, Geoff Davis, who joined Microsoft Research in the late 90's, "scholars effectively donate their intellectual property to journals by signing over the copyright to their articles, all to foster the dissemination of scientific knowledge. 'A lot of the ideals of academia are close to those of the open-source movement', he said." While there may be many faculty who fear to reveal too much of their research to others, it is in their and Dartmouth's best interest to try to convince them otherwise.

3.3 Software Ownership

An issue arose at the last CoC meeting concerning the ownership of computer code produced by the College, but paid for (at least in part) by funding from a grant. This presents an interesting dilemma.

As mentioned before, such code should be in the public domain for reasons connected to verifiability of experiments, but let's take another point of view. Suppose that ownership of the code is given to the person whose grant paid for it. Recall there is some ambiguity here since the College subsidizes the cost of development of this code. Now suppose that the College is called upon to produce similar code for someone else. They cannot use the code they wrote, because of ownership/licensing issues, so what should they do? Think of another

way to accomplish the same task? Negotiate for access? It is clear that the solution is that all such code should be Open Source. This not only eliminates all discussion of ownership claims, but it is entirely possible that the code will be of use to others, and may in fact be improved to the later advantage of computing services or previous and future clients.

3.4 Computer Fluency

The term "computer fluency" was coined in the report of the U.S. National Research Council's Committee on Information Technology Literacy (1999), *Being Fluent with Information Technology*.

The authors of that report [4]

> conclude that a computer literacy curriculum focusing on skills alone is insufficient. The ideal curriculum, they argue, would equip students with computer fluency, a 'robust understanding of what is needed to use information technology effectively across a range of applications'. ... One can conclude that colleges and universities can well serve the goals of computer literacy education by moving to a Linux standard. We should teach the concepts of operating system and office software usage, and there is no reason to use expensive, commercial products for this purpose. At higher curricular levels, colleges and universities are arguably under a positive obligation to move away from closed source software and proprietary computing infrastructures. Increasingly, it is not only scientists who must understand the details of operating systems and computing networks; advanced research in virtually every field of scholarship inevitably requires the type of intermediate to advanced understanding of information technology that was formerly possessed only by computer science graduates. In this context, open-source operating systems and networking infrastructures offer a significant advantage: they are open to dissection, analysis, and scrutiny in ways not possible with closed-source architectures.

4 Positive Aspects

- By adopting an Open Source philosophy and openly distributing our research, our source code, and our course materials we are not only raising the visibility of what Dartmouth does, but increasing our chances of meaningful interaction and collaborative efforts. The prospect of frequently reinventing the wheel is greatly reduced.
- Using Open Source software has a profound effect in any courses which discuss algorithms to accomplish a task. Applications can be dissected, analyzed, modified and tested which gives real-world significance to the contents of these courses.
- Investigating the effort of the library conversion at ASU-West may lead to significant savings for the library. In the first place, Linux is much friendlier to older hardware,

perhaps adding a year or two to the useful life of the computers. Second, running rootless systems further diminishes the load on the CPU with lightweight tasks running locally and computationally intensive ones (if they exist) run remotely.

- Support of Linux as perhaps the default platform for students would have many positive effects. Aside from serving itself as the preeminent example of Open Source software, and providing a robust desktop environment with a full complement of software applications for students, there is likely a tremendous reduction in costs associated to virus protection and removal. Student machines become infected and these infections spread rapidly leading to DOS realities for all of us on the campus. Mail servers and even all network connectivity have been shutdown repeatedly because of viruses. When one infestation is irradicated, students take their machines home only to bring them back with a new crop of viruses. The costs of dealing with these problems are significant.

  Local mirrors of the desired distribution(s) can be maintained on campus making somewhat simpler security and software updates to student machines.

5 Phased implementation

The following are a few benchmarks by which to mark our progress of adoption of an Open Source initiative.

5.1 Open File Formats

- All administrative staff use TEXT for files which might require modification, PDF for documents for which no modification is intended. Any open file format will do; let's just have a policy.

- All documents archived on the web or in department records should be in an open file format.

- All electronic submissions of documents, either by students or faculty should be in an open file format.

- HTML is also an excellent document format and could be used in conjunction with database management to streamline numerous administrative tasks. A good example is the annual Faculty Record Supplement requested by the Dean of the Faculty Office. This is a Word document, which presumably people either print and write on, or edit in Word. Well, ok, some us have it converted to TEX or another open file format. What happens to these forms? Presumably, someone in the Dean's office looks at them, and makes some decision about salary increments based upon them. But what else could they be for? How many papers and books were published by the faculty this year? How many presentations were given? How many reading courses or honors theses were given or supervised? Given the mounds of paper the DOF office collects,

a person could sift through all the reports and distill these facts, but this is so much more simply accomplished by means of an HTML form which feeds a database. All the details would still be there, the results will be uniformly readable, and data analysis is trivial.

5.2 OpenOffice

For at least Linux and Windows platforms, push the use of OpenOffice (or other Open-Source alternatives) instead of Microsoft Office to produce the TEXT and PDF formats for assignments.

Pushing OpenOffice for Mac OSX is not generally recommended at this time, as there is no "native" version. On the other hand, PDF is the native language of OSX, so any application should be able to produce PDF.

5.3 Blackboard

Faculty encouraged to make their sites as open as possible, and more faculty encouraged to develop course web sites.

5.4 Open Source in Courses

Encourage the use of Open Source software in courses. There will of course be exceptions where proprietary software is the only reasonable alternative, but the emphasis should be to use Open Source whenever possible.

5.5 Computing Services

All software developed by the College should be Open Source. A SourceForge repository of Dartmouth software would be great.

5.6 Linux Support

Begin a serious discussion of Linux, not only as a supported platform on campus, but perhaps the dominant platform for students and the library.

Aside from the previously mentioned advantage regarding viruses, such support would begin to move the College back towards one generally supported platform. This would eliminate the need for cross-platform interoperability of software, and Linux is of course the native environment in which most Open Source software is developed.

For example, both OpenOffice and Evolution (a Linux-based PIM) would be immediately available.

6 Costs

Note that this is a very short section. While there certainly will be significant costs associated to adopting an Open Source perspective, those costs must be carefully weighed against the benefits, some of which have been outlined above.

Simplisticly, an Open Source perspective is like being environmentally conscious. We buy environmentally friendly products or at least give them very serious consideration not because they are cheaper, but because they are made by companies trying to do the right thing. We should be doing the same thing within the institution, and this includes both the products we consume, as well as those we produce.

References

[1] SourceForge.net Sitewide update: March 31st, 2004

[2] Chronicle of Higher Education, The 'Open-Source Movement' Turns Its Eye to Science, November 5, 1999 (`http://chronicle.com/free/v46/i11/11a05101.htm`).

[3] ASU West Library Linux Conversion Complete, `http://linuxpr.com/releases/6172.html`, Aug 26, 2003

[4] Linux in Higher Education: Open Source, Open Minds, Social Justice,March 02, 2000, `http://www.linuxjournal.com/article.php?sid=5071`