Ingredients for an Open Source Science

By: Ana Nikolic
Student #: 0451950
ARTSCI 3CF3
Dr. Seymour

Introduction

The first seeds of collaborative software development were planted by computer scientists at universities like MIT when they wrote the first computer programs in the 1950s and 1960s. While the idea of open source software itself is nothing new, it really only hit its stride in the mid-1980s and early 1990s. The first major event in the popularization of open source was the inception of the GNU operating system (a UNIX clone) in 1984 and the subsequent creation of the GPL (General Public License) (Free Software Foundation 2007a). The second push came courtesy of Linus Torvalds, who began working on an operating system kernel based on the free *MINIX* kernel in 1991, with no other aspirations than that of playing around with his new 386 (Torvalds 1992). Over a period of a few years, Torvalds was joined by a small army of volunteer programmers and testers who helped with the development of his new system, which eventually joined with the GNU operating system to become what we know today as Linux (Free Software Foundation 2007). With the growth of the Internet and success of Linux, open source rapidly transformed from a fringe concept into a legitimate development ideology. Today, SourceForge, a popular repository for source code for open source projects has 164,698 projects and 1,749,545 registered users[1] (SourceForge 2007).

In popular culture, open source is commonly enshrined and revered because it is 'free', and is perceived as a rebellion against proprietary software behemoths like Microsoft, who are often seen as corrupt and inefficient. But what does open source really mean? There are two

[1] The number of SourceForge users and projects is accurate as of December 15 2007, but increases on a daily basis.

competing definitions, one by the Free Software Foundation, founded by Richard Stallman, of the GNU project, the other by the Open Source Initiative, supported by Linus Torvalds. According to the Free Software Foundation, software is free "like free speech, not free beer" (Free Software Foundation 2007). That is, the ability to freely examine, modify and redistribute source is a fundamental moral and ethical right we have as users. They see free software as a moral necessity, and are adamant about licenses they support not allowing open source projects to get involved with any proprietary products (Stallman 2007). By their definition, many Linux distributions are not 'free', although they are freely downloadable and open source. The Open Source Initiative also sees open source as more than just freely available software, but is far less radical than the FSF (Open Source Initiative 2007).

Transparency is the key moralistic concern of "free software". As users of a piece of software, we should have the right to know what it is actually doing and how. There is also no reason why we should not be allowed to improve it or add additional functionality to help us be more efficient. Think of purchasing a new pair of pants: once the pants are yours, you can hem them if they don't fit, or sew on a replacement button if a button falls off without having to call the manufacturer first. Even non-expert users have the right to know what is going on under the hood, regardless of whether or not they have the knowhow to appreciate this knowledge. What is powerful about this idea of transparency is that it also applies generally to the process of scientific inquiry: science needs to be an open source discipline, as scientific discoveries affect the whole world, and thus we all should have free access to information generated by scientific initiatives. In this paper, I intend to examine the ways that open source affects the processes and frameworks of scientific inquiry: directly, through open source software projects, and indirectly, on a moralistic and philosophical level.

The Spirit of Open Source

I have described the common ways that people define open source, but what is it that makes open source so special? What is the spirit of open source and where does it lie? The seminal paper titled “The Cathedral and the Bazaar,” by Eric Raymond, a well-known software pioneer, attempts to pinpoint what distinguishes open source development processes from their ordinary counterparts. Raymond describes open source development as a bazaar: a veritable free-for-all with little overarching hierarchical structure (Raymond 1999). In contrast, traditional development structures are like cathedrals, as they are highly structured and controlled. This bazaar is populated by interested and self-motivated individuals, most of whom are in the top five to ten percent of programmers, and are driven by nothing other than their passion for the work. This self-selection, coupled with the lack of hierarchical power relationships within this model, leads to more effective and more creative team work, and creates a wider base of people who can solve problems. Such hierarchy-less development is otherwise known as “extreme programming.” However, Raymond points out that no structure, including the bazaar, is capable of creating new ideas. It is rather better suited for expansion of an existing (and interesting) idea (Raymond 1999). Studies of open source projects have shown that the distribution of developer time is massively skewed, with a few developers contributing most of the lines of code (Mockus et al 2002). Nevertheless, counting lines of code is not particularly representative, since a person who contributes an efficient twenty-line algorithm is no less valuable to the team as one who contributes 500 lines of boilerplate code.

Open source in science: Sometimes it works, sometimes it doesn’t

The ideals espoused by the open source movement have become widespread in scientific circles. Though it is difficult to trace the exact sources of open source software projects in

science, one major force in this movement has been the Perl programming language. Designed by Larry Wall in the 1980s, Perl is an open source project designed to easily accommodate a large variety of custom user-contributed modules. The Comprehensive Perl Archive Network (CPAN) stores 12703 different modules, with functions ranging from the simple and functional, such as database access and statistical analysis, to the just plain weird, including the Acme::Clouseau module, which can "transform otherwise boring prose into Clouseau-esque diatribes".[2] In the 1990s, as the genomic revolution got off the ground, the amount of genetic sequence information that was stored in various labs and online databases skyrocketed. On the bioinformatics side, it became readily apparent that there were too many diverse formats and too much overlap between software genomics labs were writing. To solve this problem, a group of scientists working on the Human Genome Project began a project known as BioPerl, which was a collection of Perl modules for the processing and analysis of sequence data, available to freely use or modify (Stajich et al 2002; Stein 2002).

While BioPerl was initially intended to reduce the redundancy of code being written in labs, it has been used to generate several major projects, including EnsEMBL, an automated genome annotation tool (Stajich et al 2002). The success of BioPerl has been largely credited to its open source nature, because this allows it to transcend issues like intellectual property ownership and developer turnover, since there is no one particular person in charge of the entire project (Stajich et al 2002). By allowing the community to take part in the process of creation, they used what Raymond dubs Linus' Law—the idea that if you have enough people looking at a problem (in this case, methods for viewing and manipulating genomic information) every problem and bug will be readily apparent to someone in the community (Raymond 1999). The lack of divide between users, contributors and programmers is a democratizing force, allowing

[2] http://search.cpan.org/~martyloo/Acme-Clouseau-0.02/Clouseau.pm

progress to only be limited by the number of people passionate in the project, regardless of affiliation or background.

Another implication of open source has been the free availability of genomic data in online databases such as GenBank. Genomic analysis programs themselves are also free, enabling *anyone* with time and interest to analyze genomic data. The common use of open-source languages such as Perl in the genomics community led them to favour writing open source projects. This choice, coupled with the decision of the Human Genome Project to host genomic data in central databases, and the need for transparency in analysis tools, has made genome bioinformatics reliant on open source. In many ways, genomics exemplifies the qualities that make open source work: open source development, free collaboration, and free access to products of research.

But genome biology is more of an exception than the rule. In contrast to the abundance of open source projects in biology, there is a staggering lack of open source chemistry projects (Stahl 2005). This is because chemical compounds and chemical analysis have long been the domain of pharmaceutical companies, and as such are highly profitable pursuits. Furthermore, chemical analysis through methods like mass spectrometry requires expensive machinery, and upgrades of the bundled software are a major source of profit for manufacturers. If users don't need to purchase the upgrades, this negatively affects profit, putting open source projects into precarious territory. For example, MDL Chime, a commercial molecular visualization program, was based on RasMol, an open-source visualization tool, and no credit was given to the original author (Stahl 2005). Open source is generally regarded as anti-commercial, and it is inhibited by the influence and interest of large chemical and pharmaceutical companies.

Another counterexample to genomics comes from the related field of proteomics. Most current methods in proteomics rely on identification of proteins in a sample by analysis using mass spectrometers, machines that cost hundreds of thousands of dollars. Each manufacturer has its own proprietary software and file format for raw data and analysis. Attempts to standardize data, such as by introduction of shared XML formats and through initiatives like the Institute for Systems Biology's Trans Proteomic Pipeline project, have been largely unsuccessful (Quackenbush 2006). If you examine CIHR's requirements for proteomics data deposition, they ask experimenters to submit data into one of three different unrelated databases (CIHR 2007). Furthermore, raw spectra are of little use, as two of the three major search engines are closed source and have licenses that cost thousands of dollars. The presence of large corporations has discouraged open source, and thus hindered the open sharing and transparency of protein identification data.

Open source as a road to open access

Another major way that open source ideas have become pervasive in science has been through the propagation of open access. In general, there are two roads to open access: self-archiving, either at the personal or university level, which is referred to as the green road; and open access journals, termed the golden road (Harnad et al 2004). One of the most notable open access initiatives has come from the Public Library of Science (PLoS)—an American non-profit organization initiated by a group of scientists who believe that all people should have access to current scientific research (PLoS 2007). A few years ago, they circulated a petition, signed by thousands of scientists worldwide, that proposed making all scientific information and articles open access. PLoS focused their activities on journals and publishers, but other groups argued

that encouraging authors to pursue self-archiving was a better idea (PLoS 2001; Harnad 2004).[3] Namely, the results of scientific research must be made freely available online to encourage progress and improve understanding of science.

Today, many science journals, including some highly influential ones such as the Proceedings of the National Academy of the Sciences (PNAS) are open-access. But a big problem with open access journals is that they are certainly not free for the scientist: in fact, open access cost journal models may lead to increased expenses compared to traditional modes of distribution, since the publishers would likely generate substantially less income from traditional subscriptions. Because of the increasing costs to the publisher, this would translate to increased costs for the scientist, and that, in turn, would shut out scientists with less funding, such as those from developing countries, from publishing in well-known, well-read journals. Nevertheless, studies have shown that scientific articles share a fate of being "online or invisible"—that is, articles that are not available online in any form, subscription or open access, are far less likely to be cited than those that are (Lawrence 2002).

Information and library science experts tout self-archiving as an excellent solution to the access problem. However, this option is not without its own problems. First, by urging academics to post articles into locally hosted archives, data become decentralized, and are tougher for potential readers to find. While articles may be available online, they will be difficult for potential readers to find and use, especially considering that many search engines by default link to publisher websites, putting the free copies at a disadvantage (Harnad et al 2004).

The main draw towards open access for scientists is the publicity and the likelihood of getting more citations for their work—a reality that has been documented by studies comparing

[3] PLoS was not alone: many groups and consortia have produced similar statements, such as the Bethesda Statement on Open Access Publishing, the statement of the Budapest Open Access Initiative, Wellcome Trust statement, the IFLA statement, the Berlin Declaration, and the WSIS declaration, to name a few (mentioned in Harnad 2004).

open access to non-open access articles in several disciplines, finding a drastic increase in the number of citations for work that is available in open access form (Antelman 2004). Though this motivation is a rather selfish one, the outcome is still free access to scientific information for all.

Open access has been encouraged by the policy makers as well. For example, CIHR recently (as of September 2007) made it mandatory for all published work at least partly funded by a CIHR grant to be published in an open access journal or at the very least deposited into an open-access archive (such as PubMed Central) (CIHR 2007). The Wellcome Trust in the UK and the US congress have also drafted policies that urge, if not absolutely require, publicly funded work to be deposited into open access databases (Harnad et al 2004). The UN has also developed two initiatives, HINARI and AGORA, to enable open access to many medical and agricultural journals by scientists in developing countries (Chan and Costa 2005). However, these projects have their own problems, as many countries, including India, are not considered poor enough to receive the free or discounted access (Chan and Costa 2005). Also, by only targeting institutions, access to information by regular people is not improved, and so the base of competent observers does not grow. Open access is not identical to open source, as it just gives us access to the "source" and results of an experiment, but the underlying fundamental ideology is the same. Opening up scientific literature allows a greater number of people to examine the problems facing science: and if more eyes look at the problem, it's more likely that there will be someone there to whom the solution will be readily apparent.

Conclusion: Open source as a scientific philosophy

The concept of open source is commonly associated with the ideal of "extreme programming." But why not extend this concept into science itself? Why not go from "extreme programming" to "extreme science"? Scientists need to regain the ability to pursue something

they truly are passionate about and interested in, rather than what is directed by some faceless grant committee.

At first glance, this may seem to be a rather cut-and-dry proposition. But thinking about it more deeply, isn't science inherently an open-source pursuit? After all, science has always been fairly collaborative, relying on information propagation and transfer to advance society. Through conferences, papers and posters, scientists share their results with each other, working towards a common goal: a better understanding of the world around us. Furthermore, what is known as open source now was stimulated by academics working at university research labs in the 1960s. Linus Torvalds started writing Linux while he was an undergraduate student, and based his work on a teaching OS developed by another academic (Torvalds 1992).

But this assumption that open source is really just an offshoot of the same mentality and ideology as other fields of science falls short in several respects. First off, most scientific articles are *not* open source: they require costly subscriptions to academic journals or journal collections. This issue is why the open access movement, an offshoot of open source, has been so influential. The goal of open access is to make scientific research more open source. Again, however, open access is not a cure-all, as the publication costs for many journals are very high, and can be prohibitive for scientists working in developing countries. Also, language barriers and disparities in Internet access are further barriers to successful dissemination of scientific knowledge throughout the world.

There is also a distinct disparity between the difficulty of properly replicating a molecular biology experiment, for example, when compared with the relative simplicity of typing "sudo make" and "sudo make install" at a computer terminal. If the compile works and the program does what it should, that's proof enough in open source. Academic science is often far less

simple to replicate, requiring a vast array of costly, complicated instrumentation, as well as sensitive or difficult to obtain samples of materials in order to replicate particular results.

Yet another problematic issue is the scientific equivalent of what open source advocates label proprietary software: privately funded research, such as that conducted by military and pharmaceutical organizations. Who knows what's churning away inside the labs at Pfizer and Astra Zeneca? And on a more superficial level, yet more research is at least partly funded by corporations that may have hidden agendas. A few years ago I worked in a human nutrition laboratory, and saw articles, funded by Nestle, advocating the use of formula over breast milk. This lack of transparency is as problematic in science as in software, and is exactly what the open source movement has spent the last twenty years fighting against.

All science is essentially "open source" and collaborative, but in modern society, the collaborative nature of scientific progress is often overshadowed by individual self-interest and monetary concerns. The idea of extreme open source science is one that has been realized in a variety of ways at a variety of levels. First, large bioinformatics projects such as BioPerl are both "extreme science" and "extreme programming", as they are examples of collaboration between large groups of individuals working towards a common, collaborative goal. The Human Genome Project was another example of an open source scientific project, and one of global importance and relevance. In fact, it is because of the success of that project that we can look upon Open Source Science as more than just a lofty ideal, but something that can be realized and applied to science as a whole.

Recent initiatives in this field include the Tropical Disease Initiative, which is an open source science initiative for open source drug development (Maurer et al 2004).[4] By making drug

[4] Notably, PLoS has recently (as of October 2007) added another journal to its roster, called PLoS Neglected Tropical Diseases, which also aims to deal with the same issues as the TDI.

development, which is a traditionally commercial pursuit, into an open source project, we can break down some of the barriers that are preventing new cures from reaching those most in need. Funding predominantly exists for the diseases that are most relevant to the developed world, and so many of the plagues that affect a large proportion of the world's population are largely ignored in favour of killers like cancer and heart disease.

And open source science can go beyond interested benevolent academics—it can also allow unaffiliated individuals to pursue scientific problems. For example, a few years ago, the daughter of Hugh Rienhoff, a former physician turned businessman, was born with a mysterious birth defect. While Rienhoff may be more powerful and knowledgeable than the average individual, he approached his daughter's condition as a scientific problem, purchasing a PCR machine, isolating DNA from her blood, and then sending it for sequencing. He is searching for the genetic basis of this condition, without funding or permission from government or even private investors, but really for his own sake (Maher 2007). Science can matter to individuals, and by allowing those who are passionate for science to pursue their interests, we can not only employ Linus' Law, but also make science a purer and more affective discipline.

Another caveat that we need to keep in mind, which has been suggested by Eric Raymond, is that collaboration is powerful and can help ideas grow, but it is not necessarily the best medium for generating those initial sparks of brilliance (Raymond 1999). In science, sparks of brilliance are often a consequence of a large base of prior knowledge. If this knowledge is freely available, then the chance of another Newton or Einstein coming out of the woodwork and solving one of nature's great mysteries, or at least providing an interesting new idea, increases. Scientific pursuits are very involved and time-consuming, which is why open collaboration is often necessary. But we must be realistic: not every SourceForge project becomes Linux or

Firefox, and similarly, not every attempt at scientific collaboration will produce a Nobel Prize. Larry Wall attributes the success of Perl in part to its successful establishment of a culture associated with the language (Wall 1999). [5] This would be an interesting way to begin: by cultivating an open, freely accessible culture of scientific inquiry and advancement. It is true that we cannot force progress to happen, but we can at least take the spirit of open source, and transplant it back where it came from: into the domain of science.

[5] Larry Wall is renowned for his wit, which has contributed to the popularity of Perl. All documentation is written in a friendly tone, with jokes, as is the language itself: to read text is to "chomp", references can be "blessed", etc.

Works Cited

Antelman, K. 2004. Do open-access articles have a greater research impact. *College & Research Libraries* 65, no. 5: 372-382.

Canadian Institutes of Health Research Government of Canada. Policy on Access to Research Outputs - CIHR. 20071129. http://www.cihr-irsc.gc.ca/e/34846.html (accessed December 16, 2007).

Chan, L., and S. Costa. 2005. Participation in the global knowledge commons. *New Library World* 106, no. 1210/1211: 141-163.

CPAN. http://www.cpan.org/ (accessed December 16, 2007).

Free Software Foundation. The Free Software Definition. *The Free Software Foundation.* http://www.gnu.org/philosophy/free-sw.html (accessed December 16, 2007).

Harnad, S., T. Brody, F. Vallières, L. Carr, S. Hitchcock, Y. Gingras, et al. 2004. The Access/Impact Problem and the Green and Gold Roads to Open Access. *Serials Review* 30, no. 4: 310-314.

Lawrence, S. 2001. Online or invisible. *Nature* 411, no. 6837: 521.

Maher, B. 2007. Personal genomics: his daughter's DNA. *Nature* 449, no. 7164: 773-6.

Maurer, S. M., A. Rai, and A. Sali. 2004. Finding cures for tropical diseases: Is open source an answer. *PLoS Medicine* 1, no. 3: 183-186.

Mockus, A., Roy Fielding, and J. Herbsleb. 2002. Two Case Studies of Open Source Software Development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology* 11, no. 3: 309-346.

PLoS. Public Library of Science: Read the Open Letter. http://www.plos.org/about/letter.html (accessed December 1, 2007).

PLoS. Public Library of Science: Core Principles. http://www.plos.org/about/principles.html (accessed November 27, 2007).

Quackenbush, J. 2006. Standardizing the standards. *Mol Syst Biol* 2, no. 2006.0010.

Raymond, E. 1999. The Cathedral and the Bazaar. *Knowledge, Technology, and Policy* 12, no. 3: 23-49.

Sourceforge. http://www.sourceforge.net (accessed December 15, 2007)

Stahl, M. T. 2005. Open-source software: not quite endsville. *Drug Discovery Today* 10, no. 3: 219-222.

Stajich, J. E., D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, et al. The Bioperl Toolkit: Perl Modules for the Life Sciences. *Genome Research*.

Stallman, Richard. Why "Open Source" misses the point of Free Software. http://www.gnu.org/philosophy/open-source-misses-the-point.html (accessed December 16, 2007).

Stein, L. 2002. Creating a bioinformatics nation. *Nature* 417, no. 6885: 119-120.

Torvalds, Linus. 29 Jan 1992. “Re: LINUX is Obsolete.” Online posting. Newsgroup comp.os.minix. Usenet. (accessed 10 Dec 2007)

Wall, L. 1999. The Origin of the Camel Lot in the Breakdown of the Bilingual Unix. *Communications of the ACM* 42, no. 4: 40-41.