From iEMSs 2006 Wiki

W13b: Paper

Building a Community Modeling and Information Sharing Culture

Alexey Voinov, Raleigh R. Hood, John D. Daues, Hamed Assaf

Abstract

Free and open exchange of information in research endeavors is beneficial and can lead to much more rapid advances and important discoveries that might otherwise take much longer to achieve. Nevertheless, exchange of information is still restricted by patent law, as well as by institutional, cultural and traditional hurdles that create protective barriers hindering the free flow of this valuable commodity. We believe that one of the greatest challenges we face in creating a new open research paradigm will be building the community modeling and information sharing culture. How do we get engineers and scientists to put aside their traditional modes of doing business that discourage free and open exchange of data and ideas? How do we provide the incentives that will be required to make these changes happen? How do we get our colleagues to see that the benefits of sharing resources far outweigh the costs? We argue that timely sharing of data and information is not only in the best interest of the research community, but that it is also in the best interest of the scientist who is doing the sharing.

Keywords

Open source, blogs, research, information, General Public License, open data, open education, knowledge sharing, data sharing, idea sharing

Main Points

- By copying information from sources and distributing it to new destinations we do not lose information at the sources. Potentially we can only benefit from sharing information.
- The open source paradigm provides an example of information sharing that can be readily applied to modeling.
- Collaborative open source modeling still has limited application. There are cultural, traditional, institutional and bureaucratic reasons for that.
- The wide advent of Internet and web applications creates a new environment for information sharing that is likely to change the standards for academic success evaluation and promote a more collaborative and unified research field.

Introduction

Much of human creativity is geared towards moving energy and materials rather than information, even though information has become another crucial component of human welfare and livelihood. Information, unlike energy and materials, is not subject to conservation laws. By copying information from sources and distributing it to new destinations we do not lose information at the sources. This is what is known as a non-rival goods in ecological economics (Daly, Farley, 2003). Like for gravity by using information we do not decrease the ability of others to use it. Nevertheless, exchange of information is restricted by patent law, as well as by institutional, cultural and traditional hurdles that create protective barriers hindering the free flow of this valuable commodity. In this way we are making it excludable. It is not surprising that private companies are often reluctant to share data and software because it can impact their profits in a competitive market. Unfortunately, barriers to information exchange are also significant in the academic community, where the long-standing emphasis on publication and (perhaps unwarranted) fear of misuse of released data and software, have inhibited free and open exchange. Promotion and tenure at academic institutions is still largely dependent upon the volume of peer-reviewed publications and success in securing grant and contract funds. As a result, academic scientists have little or no incentive to spend the time and effort that is required to document and disseminate their data and/or their code for the greater good of the research community. This problem is exacerbated by the fact that grant and contract funding for research rarely provides direct support for documentation and dissemination activities. The issue is particularly acute when it comes to sharing the source code of models and data analysis software, i.e., even if a scientist or engineer is amenable to sharing the code, the effort required to provide documentation to make it useful is often viewed as an insurmountable obstacle.

U.S. funding agencies clearly recognize the pressing need to enhance communication and promote open exchange of data and information among scientists and between academic and private institutions via the Internet. The National Science Foundation, for example, has initiated several new major research initiatives that are aimed at developing and/or will explicitly require this enhanced communication. These initiatives include NEON (National Ecological Observatory Network), CLEANER (Collaborative Large-Scale Engineering Analysis Network for Environmental Research), CUASHI (Consortium of Universities for the Advancement of Hydrological Sciences, Inc.), and ORION (Ocean Research Interactive Observatory Network), to name just a few. All of these initiatives embrace the idea that developing the infrastructure needed to allow free and open exchange of large volumes of data and information will be crucial for making rapid scientific advancements in the future. For example, the success of current efforts to develop earth observatories in both terrestrial (e.g., NEON) and marine environments (e.g., ORION) will be critically dependent upon the successful development of this infrastructure because these observatories will have to collect, process and disseminate large volumes of data and assimilate them into models in a timely manner.

The challenges we face in creating a new research paradigm are many. Substantial improvements in hardware (e.g., network and computing infrastructure), software (e.g., data base manipulation software and data assimilating numerical models), and a much higher level of standardization of data formats will be required. New means for carrying out real-time data processing and automated data quality control will also have to be developed. However, we believe that one of the greatest challenges we face in this endeavor will be building the community modeling and information sharing culture that will be required for success. How do we get engineers and scientists to put aside their traditional modes of doing business? How do we provide the incentives that will be required to make these changes happen? How do we get our colleagues to see that the benefits of sharing resources far outweigh the costs? We argue that timely sharing of data and information is not only in the best interest of the research community, but that it is also in the best interest of the scientist who is doing the sharing, i.e., substantial additional benefits will be derived through new contacts, collaborations and acknowledgment that are fostered by open exchange. Numerous examples attest to this fact, some of which are described below. The real challenge we face is getting our colleagues to recognize the potential benefits that can be derived from adopting a community modeling and information sharing culture. In addition, we need to dispel unwarranted fears that many scientists and engineers harbor, i.e., that they will be “scooped” if they release their data too soon or blamed if there is a bug in their code. And finally, we need to accept the fact that releasing undocumented or poorly documented software is a preferable alternative to not releasing it at all.

In the following pages we discuss the history of the open source movement, focusing primarily on software development. This movement has its origins in “hacker” culture, and it matured in the software development community as a sophisticated and efficient means for developing software. This culture has now penetrated virtually every aspect of software development and it is certainly applicable to both information and data sharing. Although the scientific community has been slow to adopt it, we believe that building the community modeling and information sharing culture among scientists will be crucial for future advancement in earth science.

Open Source and Hacker Culture

Computer programming in the 1960s and 1970s was dominated by the free exchange of software (Levy, 1984). This started to change in the 1980s when the Massachusetts Institute of Technology (MIT) licensed some of the code created by its employees to a commercial firm and also when software companies began to impose copyrights (and later software patents) to protect their software from being copied (Drahos, 2002).

Probably in protest to these developments, the open-source concept started to gain ground in the 1980s. The open-source concept stems from the so-called hacker culture. Hackers are not what we usually think they are – software pirates, vicious producers of viruses, worms and other nuisances for our computers. Hackers will insist that those people should be called “crackers”. Hackers are the real computer gurus, who are addicted to problem solving and building things. They believe in freedom and voluntary mutual help. It is almost a moral duty for them to share information, solve problems and then give the solutions away just so other hackers can solve new problems instead of having to re-address old ones. Boredom and drudgery are not just unpleasant but actually evil.

Hackers have an instinctive hostility to censorship, secrecy, and the use of force or deception.

The idea of software source code shared for free is probably best known in connection with the Linux operating system. After Linus Torvalds developed its core and released it to software developers world wide, Linux became a product of joint efforts of many people, who contributed code, bug reports, fixes, enhancements, and plug-ins. The idea gained momentum when Netscape released the source code of its Navigator, the popular Internet browser program in 1998. That is when the term "open source" was coined and when the open source definition was derived. Both Linux and Navigator (the latter now developed as the "firefox" browser under mozill.org) have since developed into major software products with worldwide distributions, applications and input from software developers.

“The basic idea behind open source is very simple: When a programmer can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.” (www.opensource.org) Motivated by the spirit of traditional scientific collaboration, Richard Stallman, then a programmer at MIT's Artificial Intelligence Laboratory, founded the Free Software Foundation (FSF) in 1985 (**http://www.fsf.org/**). The FSF is dedicated to promoting computer users' rights to use, study, copy, modify, and redistribute computer programs. Bruce Perens and Eric Raymond created the Open Source Definition in 1998 (Perens, 1998). The General Public License (GPL), Richard Stallman’s innovation, is sometimes known as “copyleft”. A form of copyright protection achieved through contract law. As Stallman describes it: “To copyleft a program, first we copyright it; then we add distribution terms, which are a legal instrument that gives everyone the rights to use, modify, and redistribute the program’s code or any program derived from it, but only if the distribution terms are unchanged.” The GPL creates a commons in software development “to which anyone may add, but from which no one may subtract.”

“Users of GPL’d code know that future improvements and repairs will be accessible from the commons, and need not fear either the disappearance of their supplier or that someone will use a particularly attractive improvement or a desperately needed repair as leverage for ‘taking the program private”. (Attorney Eben Moglen) One of the crucial parts of the open source license is that it allows modifications and derivative works, but all of them must be then distributed under the same terms as the license of the original software. Therefore, unlike simply free code that could be borrowed and then used in copyrighted, commercial distributions, the open source definition and licensing effectively makes sure that the derivatives stay in the open source domain, extending and enhancing it.

The GPL prevents enclosure of the free software commons and creates a legally protected space for it to flourish. Because no one can seize the surplus value created within the commons, software developers are willing to contribute their time and energy to improving it. The commons is protected and stays protected. The GPL is the chief reason that Linux and dozens of other programs have been able to flourish without being privatized. The Open Source Software (OSS) paradigm can produce innovative, high-quality software that meets the needs of research scientists with respect to performance, scalability, security, and total cost of ownership (TCO). OSS dominates the Internet with software such as Sendmail, BIND (DNS), PHP, OpenSSL, TCP/IP, and HTTP/HTML. Many excellent applications also exist including Yahoo, Google, Apache web server, Mozilla Firefox web browser, the OpenOffice suite, and the GNU/Linux operating system (Wheeler, 2005).

OSS users have fundamental control and flexibility advantages. For example, if one were to write a model using ANSI standard C++ (as opposed Microsoft C++), one could easily move the code from one platform to another. This may be convenient for a number of reasons, from simply a preference from one developer to another, to moving from a desktop PC environment to a high performance computing (HPC) environment. Open Standards, which are publicly available specifications, offer control and flexibility as well. Examples in science include Environmental Markup Language (EML) and Virtual Reality Markup Language (VRML). If these were proprietary, use would be likely limited to one propriety application to interface with one proprietary format or numerous applications, each with its own format. One need only imagine the limitations on innovation if commonly used protocols like ASCII, HTTP, or HTML were proprietary. To organize this growing community the Open Source Development Network (OSDN) (**http://www.osdn.com**) was created. Like many previous open source spin-offs, it is based on the Internet and provides the teams of software developers distributed around the world with a virtual workspace, where they can discuss their ideas, progress, bugs, share updates and new releases. The open source paradigm has become the only viable alternative to the copyrighted, closed and restricted corporate software.

What underlies the OSS approach is the so-called “Gift culture” and “Gift economy” that is based on this culture. Under Gift Culture you gain status and reputation in it, not by dominating other people, nor by being special or by possessing things other people want, but rather by giving things away. Specifically, by giving away your time, your creativity, and the results of your skill. We can find this in some of the primitive hunter-gatherer societies where a hunter's status was not determined by how much of the kill he ate, but by what he brought back for others. One example of a gift economy is the potlatch, which is part of the pre-European cultures of the Pacific Northwest of North America. In the potlatch ceremony, the host demonstrates his wealth and prominence by giving away possessions, which prompts participants to reciprocate when they hold their own potlatch. There are many other examples of this phenomenon. What is characteristic of most is that they are based on abundance economies. There is usually a surplus of something that is easier to share than to keep for yourself. There is also the understanding of reciprocity that by doing this people can lower their individual risks and increase their survival.

In hunter-gatherer societies, freshly killed game called for a gift economy because it was perishable and there was too much for any one person to eat. Information also loses value over time and has the capacity to satisfy more than one. In many cases information gains rather than loses value through sharing. Unlike material or energy, there are no conservation laws for information. On the contrary, when divided and shared, the value of information only grows. The teacher does not know less when he shares his knowledge with his students. While the exchange economy may have been appropriate for the industrial age, the gift economy is coming back as we enter the information age.

It should be noted that the community of scientists, in a way, follows the rules of a gift economy. The scientists with highest status are not those who possess the most knowledge; they are the ones who have contributed the most to their fields. A scientist of great knowledge, but only minor contributions is almost pitied - his or her career is seen as a waste of talent. But in science the gift culture has not yet fully penetrated to the level of data and source code sharing. This culture has been inhibited by an antiquated academic model for promotion and tenure that is still prevalent today that encourages delaying release of data and source code to ensure that credit and recognition are bestowed opon the scientist who collected the data and/or developed the code. This model (which was developed when data were much more difficult to collect and analyze and long before computers and programming existed) no longer applies in the modern scientific world where new sensor technologies and observing systems generate massive volumes of data and where computer programs and numerical models have become so complex that they cannot be fully analyzed or comprehended by one scientist or even small teams.

Knowledge sharing and Intellectual Property Rights

The concept of intellectual property rights and the enactment of laws to protect them were first formalized in the Statute of Anne that was passed by the British Parliament in the early 18th century in an attempt to stem the rapid rise in unauthorized printing of books facilitated by the advent of affordable and efficient printing technology (Tuomi 2004). Formally, an intellectual property (IP) is a knowledge product that could be an idea, a concept, a method, an insight or a fact that is manifested explicitly in a patent, copyrighted material or some other form, where ownership can be defined, documented, and assigned to an individual or corporate entity (Howard 2005).

Although the concept of public domain was implicitly considered by the Statute of Anne, it was clearly articulated by Denis Diderot who was retained by the Paris Book Guild to draft a treaties on literary rights. In his “Encyclopedie”, Diderot advocated the systemic presentation and publication of knowledge of all the mechanical arts and manufacturing secrets for the purpose of reaching the public at large, promotion of research and weakening the grip of craft guild on knowledge (Tuomi 2004). With these pioneering ideas, Deidert set the stage for the evolvement of public domain, which includes non-exclusive IP that is freely, openly available and accessible to any member of the society.

Public domain and exclusive IP rights represent the two extremes in IP regimes, with the former providing a free sharing of knowledge and the latter emphasizing the rights of owners in limiting access to their knowledge products. Since the inception of the concept of intellectual property rights, it was argued that protecting these rights provide adequate compensations for owners and encourage innovations and technological development.

However, historical evidence and published research does not support this claim and points to lack of concrete evidence that confirms these claims (National Academy of Engineering 2003). Also increasingly many technological innovations were the result of collaborative efforts in an environment that promotes non-exclusive intellectual rights. Although most of these efforts are mainly those in the software development domain, e.g. development of Linux, it is interesting to note that the tremendous growth and development in the semi-conductor industry is mainly attributed to the highly dynamic and connected social networks of the Silicon Valley in 1960s, which was regarded as a public domain region, since information and know-how were freely shared among its members.

In the world of business, preservation of exclusive IP rights is seen as a necessity to maintain competitive edge and protect expensively obtained technology. Patents that were drsigned to stimulate innovation, are now having the opposite effect, especially in the software industry. As Perens describes: "Plagued by an exponential growth in software patents, many of which are not valid, software vendors and developers must navigate a potential minefield to avoid patent infringement and future lawsuits" (Perens, 2006a). The big corporations seem to solve the problem by operating in a "detente" mode: accumulating huge numbers of patents themselves they become invulnarable to claims from similar players. Another company will not sues them because then they will sue that company. However now we see that whole companies are created with the sole purpose of generating profit from patents. These “patent parasitea” make no products and derive all of their income from patent litigation. Since they make no products, the parasites are themselves invulnerable to patent infringement lawsuits, and can attack even very large companies without any fear that those companies will retaliate. One of the most extreme and ugly methods is known as patent farming: influencing a standards organization to use a particular principle covered by a patent. In the worst and most deceptive form of patent farming, the patent holder encourages the standards organization to make use of a principle without revealing the existence of a patent covering that principle. Then, later on, the patent holder demands royalties from all implementers of the standard. (Perens, 2006b).

Certainly these patent games are detrimental for small businesses. According to the American Intellectual Property Law Association, software patent lawsuits come with a defense cost of about $3 million. Even before the case could be fully heard, a single patent suit would bankrupt a typical small or medium-size applications developer, let alone an open-source developer (NewsCom, 2005). The smaller patent holder simply cannot sustain the expense of defending himself, even when justified, and is forced to settle and license his patents to the larger company. Besides the open source community is constantly under the threat of major attacks from large corporations. There is good reason to expect that Microsoft will soon be launching a patent-based legal offensive against Linux and other free software projects (NewsForge, 2004).

Unfotunately, universities are increasingly seeking to capitalize on knowledge in the form of IP rights. However, only a few of these universities are generating significant revenues from licensing IP rights (Howard 2005). This equally applies to individual researchers who may seek protection of findings. This clearly indicates that there is generally less value in blocking information and knowledge for the benefit of patenting.

Howard (2005) reports that research conducted by the Association for Institutional Research in the United States (Owen-Smith, Jason & Powell 2000) shows a marked differences in how researchers from different disciplines perceive IP rights and the prospect of patenting. Physical scientists from natural and engineering expect less personal gain from patent royalties, favor non-exclusive license arrangements where they rely more on providing service or consultancy and are less concerned about identifying the proper IP license. On contrast, life scientists expect more personal gain from patent royalties, favor exclusive licensing arrangements and are more concerned about protecting IP. The only reasonable explanation that comes to mind is that over time there were so many more patents issued in the physical and engineering domains that a certain saturation level may be approaching, while patenting is still relatively new to the life sciences.

Software Development and Collaborative Research

Just as public domain and exclusive IP rights represent the two extremes in IP regimes, the software development process can occur in one of two ways, either the “cathedral” or the “bazaar”. The approach of most producers of commercial, proprietary software is that of the cathedral, carefully crafted by a small number of people working in isolation. This is the traditional approach we also find in scientific research. Diametrically opposed to this is the bazaar, the approach taken by open source projects. Open source encourages people to freely tinker with the code, thus permitting new ideas to be easily introduced and exchanged. As the best of those new ideas gain acceptance, it essentially establishes a cycle of building upon and improving the work of the original coders (frequently in ways they didn't anticipate). The release process can be described as release early and often, delegate everything you can, be open. Leadership is essential in the OSS world, i.e., most projects have a lead who has the final word on what goes in and what does not. For example, Linus Torvalds has the final say on what is included in the kernel of Linux. In the cathedral-builder view of programming, bugs and development problems are tricky, insidious, deep phenomena. It takes months to weed them all out. Thus the long release intervals, and the disappointment when long-awaited releases are not perfect. In the bazaar view, most bugs turn shallow when exposed to a thousand co-developers. Accordingly you release often in order to get more corrections, and as a beneficial side effect you have less to lose if a bug gets out the door.

It is clear that the bazaar approach can work in general scientific projects and in modeling applications in particular. Numerous successful examples, especially in earth system modeling, attest to this fact. But we must also recognize that there is a difference between software development and science, and that software engineers and scientists have different attitudes about software development. For a software engineer, the exponential growth of computer performance offers unlimited resources for the development of new modeling systems. Models are therefore viewed by engineers as just pieces of software that can be therefore built from blocks or objects, almost automatically and then connected over the web and distributed over a network of computers. It is simply a matter of choosing the right architecture and writing the appropriate code. The code is either correct or not, either it works or crashes. Not so with a scientific model. Rather, most scientists consider that a model is useful only as an eloquent simplification of reality that needs profound understanding of the system to be built. A model should tell us more about the system, than simply the data available. Even the best model can be wrong and yet quite useful if it enhances our understanding of the system. Moreover, it often takes a long time to develop and test a scientific model.

As a result of this difference in point of view and approach, we tend to see much more rapid development of new languages, software development tools and open code and information sharing approaches among software engineers. In contrast, we see relatively slow adoption of these tools and approaches by the research modeling community. This is in spite of the fact that they will undoubtedly catalyze more rapid scientific advancements. As web services empower researchers, the biggest obstacle to fulfilling this vision of free and open exchange among scientists will be cultural. Competitiveness and conservative approaches will always be with us, but developing meaningful credit for those who share their data and their code will be essential in order to changes attitudes and encourage the diversity of means by which researchers can contribute to the global academy. (www.nature.com/nature Vol 438 | Issue no. 7068 | 1 December 2005, p.531. Let data speak to data). It is clear that a new academic model that promotes open exchange of data, software and information is urgently needed. Fortunately, the success of the open source approach in software development has instigated researchers to start considering similar shared open approaches in scientific research. Numerous collaborative research projects are now based on the internet communications and are led simultaneously at several institutions working on parts of a larger endeavor (Schweik, Grove, 2000). Sometimes such projects are open to new researchers to participate in the work. Results and credit are usually shared among all the participants. This trend is being fueled by the general trend of increasing funding for large collaborative research projects, particularly in the earth sciences.

Open Source Software vs. Community Modeling

The recent emergence of open source model development approaches in a variety of different earth science modeling efforts (which we refer to here as community modeling) is an encouraging development. Although the basic approach is the same, we can also identify several aspects of research-oriented community modeling that distinguish it from and open source software development. For example, there have been a number of successful community modeling projects (Table 1). However, unlike most of the open source software development efforts, these have been blessed by substantial grant and contract support (usually from federal sources), and exist largely as umbrella projects for existing on-going research. To what extend these projects are truly open to the wider community is an open question, i.e., it is not clear how new participants get involved (there are no guidelines for this on the existing web sites).

Table 1.

Name	Web site and players	Scope	Projects
CMAS Community Modeling and Analysis System	**http://www.cmascenter.org/** Funding - US EPA, Lead - Carolina Environmental Program at the University of North Carolina at Chapel Hill	Development of Air Quality and Meteorological models, extensions of the Models-3/CMAQ. Outreach, user-support	CMAS-Supported Products: Community Multiscale Air Quality (CMAQ) Modeling System, Meteorology Chemistry Interface Processor (MCIP), Sparse Matrix Operator Kernel Emissions (SMOKE), System Package for Analysis and Visualization for Environmental (PAVE), data Input/Output Applications Programming Interface (I/O API), MM5 Meteorology Coupler (MCPL), Multimedia Integrated Modeling System (MIMS)
ESMF Earth System Modeling Framework	**http://www.esmf.ucar.edu/**	High-performance, flexible software infrastructure for use in climate, numerical weather prediction, data assimilation, and other	Earth science applications
CCSM Community	**http://www.ccsm.ucar.edu/** - NCAR	Global atmosphere	Working Groups: Atmosphere

Climate System Model		model for use by the wider climate research community	Model, Land Model, Ocean Model, Polar Climate, Biogeochemistry, Paleoclimate, Climate Variability, Climate Change, Software Engineering
CSTM National Community Sediment-Transport Model	**http://woodshole.er.usgs.gov/** project-pages/sediment-transport/ - Woods Hole	Deterministic models of sediment transport in coastal seas, estuaries, and rivers	CTSM modules implemented in ROMS and FVCOM hydrodynamic models. Regional applications: Massachusetts Bay, Hudson River, Adriatic Sea
CCMP Chesapeake Community Model Program	**http://ccmp.chesapeake.org** - Chesapeake Research Consortium	Estuary, river and watershed modeling for water quality in the Chesapeake Bay	Baywide Hydrodynamic models: Quoddy, ROMS, POM, Biogeochemical models: CH3D_biowp, Larvae tracking IBM: CBOLT, Watershed: CBP-HSPF and V5 data
WATer and Environmental Research Systems (WATERS) Network	**http://www.cuahsi.org/** **http://cleaner.ncsa.uiuc.edu/home/**	Hydrologic sciences, complex, large-scale environmental systems, education, outreach, and technology transfer	CUAHSI Consortium of Universities for the Advancement of Hydrologic Science, CLEANER Collaborative Large-scale Engineering Analysis Network for Environmental

			Research

In general, in community modeling there is usually a much smaller number of participants because the research community is much smaller and more specialized than broad field of software development. Because the pool is smaller it may be harder to find the right people, both in terms of their skills and their willingness to collaborate within an open modeling paradigm. Similarly, there is generally a much smaller number of users of open source research-oriented models, which may be very specialized and usually require specific skills to use. This is mostly because scientific models are very often focused on simulating a specific phenomena or addressing a specific scientific question or hypothesis, and also because the scientific community is very small compared to the public at large. Along these same lines, research-oriented models are generally more sofisticated and difficult to use than software products that are developed for the public. It is certainly much harder to run a meaningful scenario with a model, than to aim your virtual gun at a virtual victim and press the "shoot" button in a computer game (though one might argue that to a large extent this difference in difficulty of use has more to do with the primitive state of the user interface of most scientific codes). It is also generally true that scientific codes require more sophisticated documentation and steeper learning curve to master. Documenting models becomes a real problem since this is not what researchers normally enjoy doing and this is rarely appreciated and funded. On the other hand it becomes a crucial part of the process if we anticipate others will use and take part in the development of our model.

Open research is also much more than open programming. As we mentioned above, software development has a clear goal, an outcome. The product specifications can be well established and designed. In contrast, research modeling is iterative and interactive. The goal oftentimes gets modified while the project evolves. It is much more a process than a product. It becomes harder to agree on the desired outcomes and the features of the product. In some respects modeling is more like an art than a science. Following this analogy, how do you get several artists together to paint one picture? This is particularly true in ecological modeling where there is no overarching theory to guide model structure and where a variety of different formulations can be used to represent a particular process. These aspects of scientific modeling actually make it highly amenable to open programming approaches, which naturally allow a high degree of flexibility. Another significant impediment to developing open research models is the lack of infrastructure, i.e., there are still few good software tools to support community research and modeling projects. Once again there is an obvious gap between software and application. There is software that potentially offers some exciting approaches and new paradigms to support modularity, data sharing, web access, or flexible organization – all the major components required for successful model integration and development. The most recent trends in software design are compared to the Lego constructor over the web (Markoff, 2006), exactly what we need for modular models. However, this is yet to be developed and applied to the modeling process, and embedded into the modeling lexicon and modeling practice.

Finally, returning to the central problem, we really need to change the traditional culture and attitudes of research scientists, i.e., promote a shift in the mindset and psychology that drives scientific research. Historically, most science has been driven by individual efforts and individual talent. Talent and ingenuity of individuals will always be critical in scientific exploration, but with the growing amount of data, knowledge and information, most of the breakthrough achievements are now produced in team efforts, where teams and teamwork rather than individuals are key. This trend is being driven to a large extent by the increasing emphasis in scientific research on large projects aimed solving complex interdisciplinary problems, e.g., like simulating and predicting the earth system response to global warming. It is becoming increasingly difficult to identify the sole individual who cried “Eureka!” and solved the problem. Even when it is done very often the recognition is biased by past success, hierarchy, and personalities. There is an obvious need for new award and credit systems that will stimulate sharing and teamwork rather than direct personal gain, credit and fame.

Open Data

In addition to the trend toward open source modeling in science, there has also been an increasing emphasis on timely data sharing and archiving to prevent loss of valuable information. To a large extent this trend is being driven by new requirements that are being put in place by many government research sponsors. For example, the National Science Foundation (NSF) now requires specific data management plans and time lines for archiving data in permanent repositories such as the NOAA National Oceanographic Data Center (NODC). Once these data are archived, they are available to anyone that wishes to use them. In addition, the trend of increased data sharing is also being driven by the rapidly increasing volumes of data that are being generated by increasingly sophisticated and automated observing systems. These include, for example, satellite probes and ground-based continuous monitoring sensors and sensor networks. Thus, our ability to collect and store large volumes of data is pushing science toward an “abundance economy”, i.e., where there is a surplus of data that cannot possibly be fully analyzed and understood by a single individual or small group of scientists. Open data sharing allows scientists to “hack” at information, i.e., extracting additional results, applying it to answer new questions and using it in other research programs that may extend far beyond the original goal of the program that generated the data.

For the open data model to provide the maximum value, all applications have to be able to use it, i.e., implementations of the open data model should be platform and application independent. For example, XML makes it possible for the same information to interact with multiple programs in multiple environments. Instead of the information being bound inseparably to one program, it can be read, processed, and stored by any number of programs. The Open Document Format (ODF), short for the OASIS Open Document Format for Office Applications, is an open document XML file format for saving and exchanging editable office documents (**http://en.wikipedia.org/wiki/Document_file_format**). The requirement that data from diverse sources can be easily shared is also driving a trend toward increasing standardization of not only data formats, but also data descriptions, i.e., the so-called metadata that allows a researcher to figure out where the data came from, how it was collected and how it is organized. Several organizations (e.g., the Open Data Foundation (**http://www.opendata.us**), the Open Data Format Initiative (**http://odfi.org/**), and the Open Data Consortium) have emerged in the last decade that are dedicated to guaranteeing the free access of citizens to public information, and making sure that the encoding of data is not tied to a single provider. The use of standard and open formats, such as netCDFand HDF, gives a guarantee of this free access, and also often necessitates the creation of compatible free software.

The issue of open data becomes especially important because modern governments generate a vast number of digital files every day, from birth certificates and tax returns to criminal DNA records. All of these documents must be retrievable in perpetuity and shared by numerous agencies and departments. As a result, governments have been reluctant to store official records in the proprietary formats of commercial-software vendors and so have already adopted an open data model by necessity (cite The Economist [9/11/03]). Scientists have been slow to adopt these kinds of standards for a variety of reasons, not the least of which is the understandable desire to retain privileged access to data that they have invested heavily in collecting, pending publication. Times are changing. As we discussed above, there are huge amounts of data that do not need to be kept behind walls. Moreover, it is now possible to make data available under a Creative Commons license (see **http://creativecommons.org/license**), where both rights and credits for the reuse of data can be stipulated, while allowing its uninterrupted access by machines. (cite www.nature.com/nature Vol. 438 | Issue no. 7068 | 1 December 2005, p.531. Let data speak to data). Unfortunately, very few scientists and academic organizations seem to be aware of this option.

Collaborative Teaching

It makes perfect sense to also consider how the open source paradigm can be used to advance education (Voinov, 2001). A web-based course could serve as a core for some joint efforts of many researchers, software developers, educators and students. Researchers could describe the findings that are appropriate for the course theme. Educators could organize the modules in subsets and sequences that would best match the requirements of particular programs and curricula. Software developers could contribute software tools for visualization, interpretation and communication. Students would be there to test the materials offered and to contribute their feedback and questions, which is essential for improvements of both the content and the form of representation.

Much can be learned from textbooks and recorded sources by the students themselves. However, a good teacher is always essential to facilitate and expedite the learning process. Borrowing from the open source experience of material development, we could also envision a community of educators who would participate in teaching a web-based course, logging into the virtual classroom to contribute to the discussions with students, to answer their questions, to grade their exercises. In this case the talents of the best teachers can be made available to the widest possible audience of students. With a sufficient number of qualified volunteers involved, this kind of education can become a free alternative to the increasingly expensive university education. In compliance with the open source definition the students educated for free would be expected to contribute in the future to this kind of free virtual education, further enhancing the community of educators. One could easily envision an Open Network for Education (ONE) set up in a way similar to the OSDN to promote and organize free open source education (along with open distribution of related tools and resources) in a variety of disciplines.

Summary and Conclusions

So how do we do it? How can we apply and extend the highly successful model of open source software development to open research modeling, data sharing and education?

- What is the “scientific” version of hacker’s culture?
- How can we make something useful beyond our small community (our gift economy)?
- How do we build a cathedral in the middle of the bazaar?

The major challenge we face in this in endeavor is overcoming the pervasive reluctance among scientists about releasing data and code for fear of getting “scooped”. This reluctance stems from the persistence of traditional modes of carrying out scientific research, i.e., science used to be driven primarily by single-investigator research, when it was much more experimental, and data were much harder to collect. Under those conditions, there is potentially great risk associated with giving away data or a model before full credit has been garnered through publication. This problem is exacerbated by the fact that pursuit of “fame” is major driver for many scientists, i.e., if you give away your data and your models too quickly then somebody else might publish them first and you will make them famous instead of yourself! Moreover, many scientists do not want to share their models and code out of fear of others finding their bugs and mistakes. It is not pleasant when somebody shows that you were wrong, especially in print. It is safer to keep your code and your data to yourself.

But the times have changed. The old rules and fears are not valid anymore in modern scientific research where we are awash in data, where collaborative, multi-investigator teams are the norm rather than the exception, and where models are becoming increasingly complex to address increasingly complex problems. In the modern world of scientific research it clearly makes sense to share data, code and ultimately credit. Unfortunately, universities tend to perpetuate old-fashioned behaviors because most still use traditional criterion for promotion and tenure, i.e. emphasizing first author publications, and success in obtaining grants and contracts. There is little top-down incentive to share. Fortunately, the funding agencies are starting to apply pressure to share data in a timely manner, and pressure to share code is likely to soon follow. Another big part of the problem is that there is a gap between the average scientist using a model that might be written in FORTRAN, for example, and more modern programming languages and approaches. More widespread adoption of open modeling languages that can be easily plugged into (and saved from) open model building frameworks would greatly facilitate open source modeling in research. It would allow scientists to take full advantage of modern open source software development tools like CVS (Concurrent Versions System, **http://www.nongnu.org/cvs/** - also an open source project), Subversion, etc. For open source modeling to become a reality in scientific research, we will need to be able to use the same or similar tools. Fortunately, movement in this direction is being facilitated by the growing need to develop modeling platforms that accept data from the web and that therefore use common standards and formats for geospatial data. Adoption of modern, open source programming and code sharing approaches and tools will ultimately make it possible to construct deeper and more complex models and solve deeper and more complex problems.

In addition to the need for developing new methods and approaches that facilitate open development and sharing of models and large volumes of data (cite Slocombe, 1993), there is also a demand for new “process methods” that refer to working with people, communities, and businesses in scientific pursuits. The development of the Internet creates new and unforeseen possibilities for moving scientific research in this direction. In a way, we no longer have to have a middleman, an intermediate agent between an individual scientist and the rest of the community or the public. In the past the only way to get the message out was to publish in journals, present at conferences, or write a book. Now anyone can publish on the web and sooner or later search engines will start picking up these findings and guiding the public towards them if they are of general interest. Of course, there are pitfalls in this trend because it can result in propagation of misinformation and bad science, but there is also tremendous benefit that can be derived from rapid dissemination and a much larger diversity of information sources. In a way we get a system that is parallel to peer review and may be considered complimentary in many respects.

Most likely, peer-review journals will reside entirely on-line – this trend is already apparent. Scientists have started sharing papers like people share music, i.e., by freely exchanging electronic reprints over the web. By analogy, perhaps a torrent/P2P application could be used to find and disseminate publications over the web. All researchers already have a collection of files on their computers that contain their own publications and perhaps papers that they have found interesting and downloaded from somewhere else. Scientists could share these libraries, rendering expensive journals obsolete. Hopefully publishing houses will be more flexible than the RIAA (Recording Industry Association of America) and MPAA (Motion Picture Association of America), the giant entertainment industry, and will adopt the new environment without waging wars and lawsuits against researchers and software developers. We already see a number of open access scientific journals on the web, such as First Monday (**http://www.firstmonday.org/**), Ecology and Society, the Living Reviews series and Scientia Marina. This is an exciting trend that is likely to grow as we move to fully electronic publications.

We already witness how research communities are organized spontaneously around certain topics, and how group initiatives similar to research projects are developed. Consider, for instance, the Oil Drum project that currently is developed at **http://www.theoildrum.com/**. A self-organized group of people who share similar views and concerns are working on various issues that interest them and that are related to the topic they chose. They are publishing data and findings on their blog for anyone to see and participate. There is an active community that is engaged in discussions, and that posts comments and questions, which further enhance and direct the research. All this is done on a totally volunteer basis. Another example is the on-line research spearheaded by Dr. Henry Niman, who analyzes the dynamics of bird-flu with a blog of his own, where volunteers can help track local press and radio reports to understand the trends of the epidemic (Recombinomics, 2006). Ridiculed by WHO and other official science (Zamiska, 2006) the results of this analysis gradually turn out to be quite well recognized in later studies of bird-flu. More recently some of the predictions of Niman are reported to be even more accurate than the official science (McNeil, 2006). Can we consider these examples as harbingers of future distributed open source research over the Internet? Unfortunately, standard methods of accounting for scientific success do not account for participation in this kind of research. However, in terms of impact and importance, we would argue that this kind of activity deserves as much recognition as the highly desired publications in some recognized peer-review journal. These standards will need to change.

We see the future of science moving strongly toward more collaborative and open research where data, code and credit are much more widely shared, and that embraces the development of this kind of self-organized and community driven research. In this new scientific era the number of hits on individual home pages, and numbers of posts on scientific blogs will become as important indicators of scientific success as the numbers of publications in “Science” or “Nature”. “In the new world-view, the universe is seen as a dynamic web of interrelated events. None of the properties of any part of this web is fundamental; they all follow from the properties of the other parts and the overall consistency of their mutual interrelations determines the structure of the entire web” (F.Capra). Clearly, we are entering an era, when the free flow of information becomes crucial to tackle the pressing problems of our future, when the complexity of the problems and associated hypotheses and data sets will require well coordinated team efforts, and when individual scientists will be best recognized and valued for their ability to contribute to the team effort, to share their knowledge, skills and ideas.

References

Daly, H., Farley, J., 2003. Ecological Economics. Island Press.

Freiburger, P. and M. Swaine, 2000, Fire in the Valley, New York, McGraw-Hill

Hippel, E and G Krogh, 2003, Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science. Organization Science, Vol. 14, No. 2, March-April 2003, pp. 209-223.

Howard, J., 2005, “The emerging business of knowledge transfer: creating value from intellectual products and services.” Canberra: Australian Government Department of Education, Science and Training.

Kipp, M.E.I, ,2005, Software and Seeds: Open Source Methods Levy, S., 1984, Hackers. Anchor/Doubleday, New York.

McNeil, D., 2006. Human Flu Transfers May Exceed Reports , New York Times, June 4, 2006.

Markoff, J., 2006. Software out there. The New York Times, April 5, 2006. **http://www.nytimes.com/2006/04/05/technology/techspecial4/05lego.html?ex=1146628800&en=d6728f2af081fb16&ei=5070**

National Academy of Engineering, 2003, “The impact of academic research on industrial performance”, National Academies Press, Washington.

Newscom, 2005. The open-source patent conundrum. **http://news.com.com/2102-1071_3-5557340.html?tag=st.util.print**

Newsforge, 2004. HP memo forecasts MS patent attacks on free software. **http://www.newsforge.com/article.pl?sid=04/07/19/2315200**

Owen-Smith, Jason & Powell, Walter, 2000, "To patent or not to patent: Faculty decisions and institutional success at technology transfer", Journal of Technology Transfer.

Perens, B., 1998, The open source definition. **http://perens.com/Articles/OSD.html**

Perens, B., 2006a. Software Patents vs. Free Software. **http://perens.com/Articles/Patents.html**

Perens, B., 2006b. The Problem of Software Patents in Standards. **http://perens.com/Articles/PatentFarming.html#2**

Research Coordination Networks in Biological Sciences (RCN) [nsf06567] URL : **http://www.nsf.gov/publications/pub_summ.jsp?ods_key=nsf06567**

Recombinomics, 2006. **http://www.recombinomics.com/**

Tuomi, Ilkka, 2004, “Knowledge sharing and the idea of the public domain”, UNESCO 21st Century Dialogues, “Building World Knowledge Societies”, Joint Research Centre, Institute for Prospective Technological Studies, Seoul, Korea.

Wheeler, D., 2005, Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers! **http://www.dwheeler.com**

Zamiska, N, 2006. Blogging biochemist tracks bird flu, but scientists remain skeptical. The Wall Street Journal, March 24-26, 2006, p.28. **http://agonist.org/20060323/niman_a_bird_flu_watcher_develops_a_following_through_the_internet**

Retrieved from http://www.iemss.org/iemss2006/wiki/index.php?n=W13b.Paper
Page last modified on October 11, 2006, at 11:30 PM