FOSTERING OPEN-SOURCE RESEARCH VIA A WORLD WIDE WEB SYSTEM

Charles M. Schweik, Ph.D.
University of Massachusetts, Amherst

J. Morgan Grove, Ph.D.
USDA Forest Service

Research support for this study was provided by the Burlington Laboratory (4454) and Global Change Program, Northeastern Forest Research Station, USDA Forest Service, the National Science Foundation-NSF Grant #DEB-9714835 - and Environmental Protection Agency-EPA Grant #R-825792-01-0.

Abstract

In recent years we have witnessed the incredibly productive power of "open-source" programming where independent software developers freely share their source code and collaborate globally over the Internet. While components of the Internet (e.g., email, FTP, etc.) have long been used as mechanisms for research collaboration, it has only been recently – since the development of the web – that the Internet as a system for research collaboration has been available to non-technical users. Even so, the free exchange of research data and products, similar to the open-source sharing of programs is still quite limited. This paper explores the question of how a web system might enhance and encourage open source modeling of land cover change and, in general, "open-source research." We discuss the concept of open-source and the creative and productive potential of open-source collaboration. We describe the foundations of open-source programming, largely in the context of Linux, and summarize lessons learned from these open-source efforts. Finally, we examine how these lessons might be applied in an open-source research setting by describing our initial efforts to establish a web system to encourage and foster open-source research and modeling of complex human-environment systems.

Public Administration & Management:
An Interactive Journal
5, 4, 2000, pp. 161-189

Introduction

The study of complex systems poses great challenges for physical, biological, and social scientists, for it requires vast amounts of data, knowledge, and human capital in order to fully understand them. Consider, for example, the challenge of understanding global environmental change. Components of the problem fall under traditional physical science disciplines such as biology, geology, geography, and atmospheric sciences. But other components also require input from social science disciplines including political science, public policy, demography, sociology, economics, and others. And research becomes even more complex when we consider the challenges brought about by the problem of multiple spatial and temporal scales of analysis.

The traditional approach to research of complex systems is through the process of grant proposals, review by peers, and publication of results in peer-reviewed outlets. These types of endeavors are usually undertaken by one or more individuals within an organization or, in some circumstances, several individuals or organizations who pool their resources, knowledge, and talents. Recently, funding organizations such as the U.S. National Science Foundation (NSF) have encouraged the study of complex human-environmental systems by awarding grants to multidisciplinary teams. (1) The hope is that through the funding of hundreds of individuals and organizations, a deeper understanding of the complexities between humans and the environment will be understood.

Recently, however, a rather remarkable new model of collaboration has appeared out of the software industry that has already proven itself highly innovative in solving complex problems. This phenomenon has recently been labeled the "open-source initiative" (www.opensource.org). Open-source programming describes the situation where programmers collaborate freely and share their intellectual property — readable versions of their programs — to others on the Internet. Several open-source endeavors have resulted in the development of highly complex but remarkably efficient and high-quality software products, with perhaps the most visible being the Linux operating system and the Apache web server (*O'Reilly and Dyson, 1998*). Interest in the open-source programming concept has now moved into the realm of programming in scientific

research (*Gezelter, 1999*). In this paper, we consider whether the model of collective action called open-source can be extended beyond programming to the study of complex systems — in this case human-environment systems — and whether it can be compatible with existing scientific processes.

The paper is divided into three main parts. First, we discuss the concept of "open-source" and the creative and productive potential of open-source collaboration. We describe how open-source practices are beginning to be adopted in scientific research, and we outline our vision of open-source research as it applies to the study of human-environment interactions. Second, we describe the foundations of open-source programming, largely in the context of Linux development, and summarize lessons learned from these open-source efforts. Third, we discuss how these lessons might be applied in an open-source research setting by describing our initial efforts to establish a web system to foster open-source research on two complex human-environment systems: the Chesapeake Bay Watershed and the Adirondack Park in New York State. We outline some of the primary challenges anticipated in initiating such a system and we describe design ideas to overcome these challenges.

The Creative and Productive Potential of Open-Source Collaboration

According to Opensource.org, the history of open-source dates back to the creation of the Unix operating system, the creation of free software on the Internet, and the culture of computer programmers called "hackers" (*http://www.opensource.org/history.html*). But the Linux history provides one, if not the best, example of this software revolution.

In 1991, Linus Torvalds, a college student and computer programmer in Finland, wanted to use the Unix operating system, but at the time securing the software and a computer platform was financially out of his reach (*Learmonth, 1997*). To solve this problem, Torvalds decided to program the core of a Unix operating system on his own. Eventually, he posted his creation or "kernel" – the core of a Unix-based operating system he called "Linux" – on the Internet for everyone to freely read, download and modify. (2) The posted programs caught the attention of several other programmers facing the same dilemma as Torvalds. They took advantage of the readable programs, made improvements to Linux, and, following

Torvalds' lead, posted their programs in readable form back on the Internet. In this way, arguably the most successful example of the "open-source" software revolution was initiated. (3) Since then, global programming collaboration has continued. Linux is now reported to be the most popular Internet-connected operating system in Europe, and is estimated as having up to 27 million users (*Opensource.org, 2000*). Linux is now seen as a legitimate threat to prominent commercial operating systems like Microsoft Windows (*The Economist, 1998; Wallich, 1999*).

The Linux phenomenon is a remarkable example of how the Internet and "virtual communities" (*Rheingold, 1993*) — in this case a global virtual community — can work collaboratively to solve a common problem. It could be argued that the Linux development effort has become one of the most productive and creative global initiatives the world has ever witnessed. In fact, it may be the first project to consciously make an effort to tap into the entire global population for assistance (*Raymond, 1999*). The cumulative creativity and rapid support when problems are discovered by this global team have produced an extremely robust, high-quality software product (*Opensource.com, 2000; The Economist, 1998*). Linux is such "clean" software largely because of the number of individuals who have worked on the problem and the natural peer review process that comes with an open-source endeavor (*O'Reilly, 1999*). Modifications to Linux are chosen because others with high levels of expertise select and use the best programs that are posted by programmers in the Linux virtual community.

The open-source programming movement has moved beyond the development of operating systems to the development of other software applications (4), and very recently there has been a push to bring open-source into the realm of scientific programming (*Gezelter, 1999; Kiernan, 1999; Wilson, 1999*). This is considered a natural fit, since scientific research is fundamentally an open-source endeavor itself through the sharing of intellectual property. But scientific research-sharing has largely been through formal publications, conferences, and education and, to a lesser degree, data. However, the sharing of data has quickly taken advantage of the Internet. For instance, there are data repositories like the one at the University of Michigan (*see* http://www.icpsr.umich.edu/archive1.html) that provide the free sharing of data collected from past research projects. But computer programs, written by researchers as part of the analytic endeavor, are typically not shared with

the research community. Recently, a small but growing number of scientific researchers have argued that this lack of sharing of the programs leaves a "cloud of uncertainty over the validity of modern, computer-intensive scientific research" (*Kiernan, 1999*). They argue that, in the age of the computer, good science cannot be achieved without the ability to verify all components—not just the published results and the data, but the computer programs used and the documented analytic stream that produced the results.

While we agree that providing the ability to validate research at every stage is vital for the scientific community to generate knowledge, we wonder still if the open-source concept, as it applies to research, could have another important *raison d'être*. Humanity faces a number of complex and daunting problems, and the Linux phenomenon reveals how the concept of open-source collaboration can swiftly address complex problems with a great deal of creativity and with extremely high-quality results —arguably better than any one organization can do alone. (5) We wonder, then, whether the open-source concept can be applied to other complex problems where scientific research is needed, and not just ones where scientific programming is required. While there are many examples we could propose, our main interest lies in enhancing research on complex environmental systems. Specifically, we are interested in initiating an open-source research effort to understand the complexities in human-environment interactions over several geographic scales.

Consider, for example, an investigation into the complex human-ecological processes that exist in an area such as the Chesapeake Bay Watershed. This is a huge watershed covering 64,000 square miles and six states (*Chesapeake Bay Program, 2000*). One question that interests many people is the impact of human activities (e.g., agriculture and industrial) in and around the Watershed and how these activities are changing the "health" of the Bay. A second broad human-environment question facing researchers involved with the global change research program is how human incentives and actions are changing the landcover (e.g., forest growth or decline) and how this affects the global carbon budget (*USDA Forest Service Northern Global Change Research Program, 2000*). These research questions are extremely complicated because they involve contributions from many physical and social science disciplines and are muddled by problems related to spatial scale.

What would be the possibilities if an open-source research community could be initiated to study human dimensions of environmental change? Imagine the potential increase in productivity if individuals and organizations interested in various components of human-environment research could be harnessed to work collectively in an open-source-like environment. Suppose, for example, researchers openly collaborated to create dynamic models of landcover change, in a fashion similar to how Linux programmers contributed to Linux development. In an open-source research situation, a major obstacle, the lack of data for modeling endeavors, might be overcome by encouraging the open sharing of various socio-economic and biophysical datasets. This would be especially helpful with problems we currently face in trying to model broad geographic regions or model across spatial scales. In addition, in an open-source setting, various models could be posted for others to review, download, apply to their own empirical or theoretical settings, build upon, and re-post. For instance, spatially explicit models could be generated using several Geographic Information Systems (GIS) layers provided by collaborating organizations. With a modeling "kernel" developed, various participants or organizations could contribute data layers that are needed to enhance the models. Rival models or different approaches to modeling could be posted and discussed, as when two similar enhancements were posted to the Linux open-source Internet repository. In this scenario, the creative and productive powers of many individuals and groups could be harnessed with a common goal, as opposed to distinct research groups working on their own. This describes what we mean by the phrase "open-source research."

It was natural for the concept of open-source collaboration to first appear in the context of programming, since computer scientists and engineers were the first to use Internet collaboration tools such as File Transfer Protocol (FTP), Telnet, and early bulletin boards. But the World Wide Web moves Internet access and collaboration to a new level. The explosion of e-commerce is just one example of how far-reaching the Internet and the Web have become. But how do we capitalize on these advances to encourage open-source research as we have just described it? How can we develop a web system that fosters open-source research studying human-environment relationships?

Open-Source Programming: Foundations and Lessons

An obvious starting point toward understanding how a research endeavor can be created is to ask the question: How does an open-source initiative begin?

A Common Problem or Need and an Open-Source Initiator

In the software industry, an open-source endeavor is fueled by the existence of a common problem or need and the collaboration process is ignited by an initiator who steps in to solve it (*O'Reilly, 1999*). As Raymond (*1999*) points out, "The best [open-source initiatives] start out as personal solutions to the author's everyday problems, and spread because the problem turns out to be typical for a large class of users." In the Linux case, the need was for a Unix operating system capable of running on a PC platform (*Learmonth, 1997*) and Torvalds was the initiator through his efforts to generate the "kernel" or core of the Linux operating system.

Interested Developers/Users and the Establishment of a Virtual Community

A virtual community is defined as a group of people who carry on public discussions long enough and with sufficient human feeling to form levels of personal relationships over the Internet (*Rheingold, 1993*). Blanchard and Horan (*1998*) identify two fundamentally different types of virtual communities: (1) physical-based and (2) interest-based. A physical-based virtual community is one with participants who live relatively close to one another, share some common interest(s), and augment face-to-face (FTF) interaction by participating in Internet communication. Interest-based virtual communities, on the other hand, are geographically dispersed. Members participate because of their shared interest in a topic and not because they live their lives within the same geographic region. In many interest-based virtual communities, participants may never interact in FTF settings – the transaction costs to meet FTF may be too high. Given that Linux programming participants are from all over the world, the Linux virtual community is an excellent example of this type of interest-based group.

The establishment of an active and interested virtual community is vital to the open-source development model. Linux has become such a

success primarily because it has an immense development and user base that has grown from the initial three or four interested programmers in 1991 to an estimated 27 million today (*Opensource.org, 2000; O'Reilly, 1999*). Some argue that Torvalds' genius is not that he was such an innovator in programming, but rather that he recognized the productive power of the growing virtual community and treated its participants not only as users but as co-developers (*Raymond, 1999*). Torvalds knew the system he was trying to build was too large and complex for him to develop alone. He needed human capital. (6) Through the Internet communication channels that linked the virtual community, Torvalds encouraged Linux users to discover problems, report them, and strive to fix them. As new programs were received he worked to make them immediately available to the community. The Linux virtual community was stimulated and rewarded continuously: they were "stimulated by the prospect of having an ego-satisfying piece of the action and rewarded by the sight of constant (even daily) improvement in their work" (*Ibid., 1999:7*).

Individual Motivations to Collaborate

The early developers of Linux faced a typical collective action dilemma. All participants interested in having a Unix operating system running on their personal computers *would* be better off if they were to cooperate and contribute to the development effort. But collective action theory suggests that, in the absence of a body (such as a government or an organization) overseeing the activity and enforcing appropriate behavior, individuals would not be expected contribute to the effort (*Ostrom, 1990, 1999b;Fountain, 1997*). The fundamental collective action puzzle in the Linux context is this: Why would a programmer decide to give up valuable intellectual time for which he or she could be getting paid and freely donate programming or testing skills to the group endeavor instead? Why would he or she want to do this when, in an open-source setting, it would be easy to shirk responsibilities and still get the posted software at any time?

The answer is in part due to the "hacker" culture (*Raymond, 2000*) and how we measure costs and benefits. If we measure these strictly by monetary standards, then there may be substantial incentive not to participate in an open-source initiative. But the idea of Linux struck a passionate cord with many programmers who wanted more from it than what the original kernel could provide (*O'Reilly, 1999*). Moreover,

Raymond (*1999*) argues that the utility function that Linux and other open-source programmers are maximizing is not a monetary one but rather one related to ego and self-satisfaction. Good programmers can easily make a good living in current economic conditions, and do so for other projects. It is generally thought that the motivation for the voluntary collaboration in open-source settings is the personal satisfaction that these programmers achieve in having their work recognized by their peers and the thrill of advancing a technology (*McHugh, 1998; Raymond, 1999*).

The Role of Social Capital

It is likely that the components described above — a pressing need, an originator with an initial product, and an interested and motivated development/user community —are necessary conditions for an open-source programming endeavor to be successful. It is an open question as to whether the existence of a virtual community is a necessary condition, but it does greatly increase the likelihood of open-source success for it reduces participation transaction costs and can increase the number of participants significantly. Lessons from open-source programming endeavors suggest that having a large number of participants will raise the likelihood that a high quality open-source product might be delivered. This is stated in what Raymond's (*1999*) now classic quote on group size and open-source (what he calls "Linus' law"): "given enough eyeballs, all bugs [computer programming problems] are shallow." In other words, the more eyes there are looking at a problem and thinking about a solution, the easier the problem will be to solve because someone or some group will come up with an elegant and robust solution.

Yet both theoretical literature on collective action and the empirical literature on open-source endeavors suggest that still another component is required for successful collaboration to occur: the establishment of social capital. "Social capital" refers to group attributes such as trust, norms, rules, and expectations that individuals collectively bring to their interactions with one another and to recurrent activities (*Coleman, 1988; Ostrom, 1990, 1992, 1999a; Putnam, Leonardi, and Nanetti, 1993; Putnam 1995a, b; Fountain, 1997*). In instances where collective action is needed, humans can become more productive than they might be individually if they can agree to work together and they establish credible commitments over the coordination of future activities that benefit the collective good (*Ostrom,*

1999a). But what types of social capital are important in open-source endeavors?

Trust is a centerpiece of the concept of social capital. In a collective action situation, the greater the level of trust between participants, the more willing they will be to cooperate and work with one another (*Blanchard and Horan, 1998*). Trust is developed over time as participants learn about each other's reliability though sequences of interactions or networks (*Putnam, 1995b*). Fortunately, trust is transitive. Smith may trust Jones even though she has had no experience working with him because Thompson trusts Jones and Smith trusts Thompson (*Fountain, 1997*). This means that a virtual community with many participants can achieve a high level of trust across the group in a shorter period of time because a history of trust can be transferred through verbal or written communication.

In the Linux setting, Torvalds played the role of group coordinator and has protected the kernel — the foundation of Linux. But in the early days of the Linux virtual community, there had to have been a reasonably high level of trust at least among the core participants. Certainly those who continued to provide high-quality enhancements would be less scrutinized when a new program was posted for others to use and test. Moreover, in the network of Linux programmers, the transmission of trust through the group about programming skills of participants undoubtedly occurred.

Shared norms are also critical in the establishment of trust between participants (*Putnam, 1995b*). There could be many types of norms in a situation, but one that is considered vital in social capital literature is the norm of reciprocity. Reciprocity implies that each participant achieves some level of balance between giving and taking in a common endeavor without having to establish a more formal *quid pro quo* arrangement (*Ostrom, 1999a*). If someone undertakes an act that benefits the group as a whole, the assumption is that others will undertake an act of comparable worth in the future (*Blanchard and Horan, 1998*). In instances where this norm is high, free riders — participants who take but never contribute in return — are gradually less trusted and could be eventually shunned by the group.

Surely in the early days of Linux development, when the number of participants was small, a degree of reciprocity existed. And there is

evidence that this continues to this day. In one Linux online forum for programmers we recently read this post: "We're building Linux because we want it and need it so if you like it too then why not contribute some of your code too." Tim O'Reilly, a publisher of open-source related literature, says that the cooperative rule of open-source endeavors is "everybody contributes so everybody benefits" (*O'Reilly, 1999*). But given the huge number of Linux programmers — McHugh (*1998*) estimates it to be in the thousands — and the rapid response of fixes and support from this huge group, the concern over someone free-riding is probably very low. However, in open-source efforts with a small number of participants, the norm of reciprocity could be very important.

Generally accepted standards are another type of norm that can be important in certain collective action situations. At a particular time and place an individual may make certain decisions or approaches on how to solve a particular problem. The particular approach taken may then establish a standard for how to handle the problem if it appears again in the future. This can lead to substantial reductions in transaction costs down the road when participants agree the convention serves the group in a positive manner (*Ostrom, 1999a*). In open-source programming efforts, standards play a vital role to the success of the endeavor in several ways. First, open programming standards allow people to make connections between new and existing software. Second, well-written documentation is critical because it eases program maintenance and it helps new programmers understand software design. Raymond (*1999*) notes that the Linux programming community has generated extremely high-quality written products — better than much of what can be found in commercial efforts. Still one other vital convention that appears to have enhanced the success of several open-source initiatives is a commitment to a modular architecture that provides an environment where new components can be added without having to change the core functions (*O'Reilly, 1999*).

Finally, an established system of rules is another component of social capital that often helps a group overcome collective-action problems (*Ostrom, 1999a*). In such dilemmas, rules are established either to distribute the costs of undertaking an activity or to distribute the benefits in an equitable fashion (*Ostrom, 1990, 1992*). Licensing is an important body of rules guiding open-source endeavors. There are several existing and evolving models (*see O'Reilly and Dyson, 1998, for more details*). Linux

follows the "GNU" general public license and the concept of "Copyleft." Copyleft gives unlimited permission to copy and modify the software. It also requires that the user freely distribute, without fee or additional license terms other than Copyleft, any complimentary source code that the user may have created. It is this rule structure that allowed Linux to flourish.

Applying Open-Source Concepts to a Scientific Research Endeavor — The Development of the "Open-Research System"

As we mentioned earlier, scientific discovery has always been an open-source venture to a large degree. Science relies on replication and without the source — the data, hypotheses, test approaches, and the results — replication is not possible (*O'Reilly and Dyson, 1998*). The computer has greatly advanced science but it has also made replication more of a challenge. This is why chemist Dan Gezelter (*1999, 2000*) and others (*see Wilson, 1999*) are calling for open-source programming in science, so others have all the information necessary to test and replicate a scientific endeavor.

Of course, replication is an important reason driving our effort to create an open-source research initiative, but as we stated earlier, there is another equally important reason. The open-source revolution, and the Linux example in particular, are exciting to us because they illustrate genuine global collective action, where ingenuity, productivity, and quality are extremely high and a solution to a very complex problem can be developed. Consequently, we now ask: Can the open-source model and the lessons described above be applied to scientific research of a complex system? More specifically, how might we develop a pilot open-source research system that addresses the complexities of human-environment relationships in the Chesapeake Bay Watershed and the Adirondack Park? Let us follow the same logic as we did above in the Linux analysis.

The Common Problem: Our Research Situation

Several years ago, a state forester we were working with asked a simple question related to suburban sprawl in a particular area of his state. His question went something like this: "I want to know where the 'development fringe' is, so I can identify the most threatened forested areas

and try and protect some of them." This is a common problem faced by many public managers in the U.S. and elsewhere. Similarly, the U.S. Forest Service's Northern Global Change Program is faced with the task of making projections of how much forest cover will exist across the northern United States in future years given various policy scenarios. From a research standpoint, this type of information is needed to better understand global change processes. From a practical standpoint this type of information is needed to address requirements established in the Kyoto protocols (*see http://www.cop3.de/*). These are just two of many environmental policy-related questions that require analysis and modeling of landcover change. Obviously, any effort to do this kind of modeling requires attention not just to the biophysical components but also to the human components.

If regional and local policymakers, researchers, and other citizens had such a modeling system for a region of interest (such as the Chesapeake Bay and the Adirondack Park or subregions within these areas) this could help them understand and address various local and regional issues much better than they could without such a system. Thus the need for a modeling system has not only a global set of interested parties (often researchers interested in broader geographic regions) but also sets of local and regional interested parties (most likely policymakers, researchers, environmentalists and citizens) who have more geographically specific research or policy-related questions. In short, given the complexity of the problem and the interest base, landcover research and modeling is a natural candidate for an open-source effort.

Initiating an Open-Source Research Endeavor

In the Linux context, the initiation of the open-source endeavor was an initial product, version 1.0 of the Linux kernel, and a conversation over the Internet between Torvalds and several other interested programmers. We are taking a similar approach to establish our open-source research concept.

We started by conceptualizing what we wanted to do, having internal discussions, and generating interest among colleagues in our own organizations. We then began branching out. We contacted several organizations that we knew had an interest in human dimensions of environmental change research and modeling and/or a specific interest in

understanding landcover change either in the Chesapeake Bay or the Adirondack Park region. At this point we were soliciting interest in the idea, gauging reactions, and beginning to establish an initial core group of interested actors. The reactions to our initiative were positive.

Next, we needed an initial product for people to begin working with – our own "open-source research kernel" for people to build upon. Our concept of open-source environmental research has several core components: (1) knowledge and availability of data; (2) knowledge of persons and organizations with various expertise; (3) knowledge of existing applicable research; (4) mechanisms to assess the quality and compatibility of these data; and (5) one or several generic modeling systems or environments.

We realized, then, that what was needed for the first three components above, was a web-based metadatabase on both social and biophysical data for the two pilot areas (Chesapeake Bay and Adirondack Park) as well as more generic information related to human dimensions of land cover change modeling. (7) The web database also must document who was actively studying the region. Our efforts over the past year have been to develop such a web metadatabase. The initial homepage is displayed in Figure 1 and the initial release of this website can be found at www.open-research.org.

Figure 1
The Initial Homepage of the "Open-Research System"

In its current design, the web system has several functions to enhance research and modeling collaboration. Under the "Submit Data" option, registered users will be able to submit metadata about data in their possession to the metadatabase for others to query. If he or she desires, the data owner is given the option to submit the data itself to our server. In this fashion, our system will act as a data clearinghouse, much like others that

have been established. (8) These functions will allow us to build an open-source research metadatabase and data repository associated with human dimensions of environmental change research. Data types that can be posted to this metadata system include: Geographic Information System (GIS) data layers, such as road networks, landcover maps, satellite images, political boundaries, etc; non-geographic data such as spreadsheets on timber prices over time; citations of relevant publications or reports; and information about a person or organization with a specific research interest or expertise.

In addition, we decided to add a new type of data — website reviews — to assist researchers in finding useful research websites. We contemplated four approaches to this submit function. One option would be to create a web review facility that is similar to what is found in the book review option at www.amazon.com. Here the reviewer, anyone with any level of expertise, is treated as anonymous. A second option would be where the reviewer could again be anyone with any expertise, but the reviewer's name is posted with the web review. The third option would be to treat the web review like a peer review process of a submitted paper to a journal, where a "web review editor" is established who requests the web review to be conducted by someone with specific expertise, but the reviewer remains anonymous. The fourth option would be a formal request by a web review editor to a reviewer with specific expertise and this reviewer's name is posted with the review in the web-database for others to see. This fourth option is much like a review in the *New York Times* book review section. For reasons of reviewer motivation, discussed further below, we chose this fourth option.

We should note that a crucial design issue was to ensure that our metadatabase system is compatible with standards established by the Federal Geographic Data Committee (*FGDC, http://www.fgdc.gov/*). This is important, in part, because there is a federal mandate requiring federal agencies like the USDA Forest Service to comply with these standards. However, for open-source research, FGDC compliance serves another purpose. An important component to building an open-source research effort will be how easily other potential collaborators can find the site and the associated data products stored on the web server. For this reason, we designed our metadatabase structure on the "metalite" PC database provided by the USGS (*http://edcnts11.cr.usgs.gov/metalite/*) and are adding some new fields to this structure to allow for the other types of records we will be

collecting (e.g., models, web reviews, publication citations). By designing this database to be compatible with FGDC standards, we intend eventually to have our server recognized as a FGDC clearinghouse server. This will allow new technologies like Mapinfo's metadata browser (*http://www.mapinfo.com/software/mdb/*) to search our server, thereby increasing the likelihood of other researchers finding our website.

Finally, users will be able to submit metadata on landcover models they have developed and also supply the actual model source code to the server if they desire. We have already identified one modeling effort related to the biophysical components of landcover change (*http://www.pnet.sr.unh.edu/index.html*) and these researchers treat their modeling software as open-source. In addition, we are working with researchers at the Center for the Study of Institutions, Population and Environmental Change (*http://www.cipec.org*) to develop a landcover change model that incorporates the human dimension into the modeling process.

The other major function the open-research system provides is a search facility for metadata and also datasets and model software that reside on the project server. We are currently programming several search mechanisms for end users. One mechanism that is currently available is a standard keyword search facility where users can search for one or more specific types of metadata (e.g., datasets, publications, models, web reviews) or search all types for keywords of interest. If related datasets, software or publications exist on the server, users will be able to download them for use at their own organizations. If the data exists elsewhere (e.g., someone else's web site) the system will return a hyperlink to the site or, at a minimum, contact information for the owner of the data. We also plan to provide at least two other approaches for searching data. One is a graphical approach, where theoretical concepts related to human-ecosystem linkages are displayed and, when the end user clicks on certain parts of the graphic, various searches are invoked.

If, after searching, users decide to download the data or model software and augment it in some way, they can then return to the post functions and post the new version to our system — thus producing an open-source type of research environment where all users are considered co-developers. In short, our initial product to initiate an open-source

research project for human dimensions of environmental change is a metadata website and clearinghouse to allow various groups to store their metadata and share their products with the broader community. The system will also provide other standard features for Internet collaboration such as email discussion groups.

Establishing Interested Virtual Communities

While we already have established an initial group of interested participants, the lessons learned from the Linux and other open-source initiatives suggest that a higher number of research innovations will occur as the virtual community increases. (9) But just what kind of community are we trying to establish?

Recall that Blanchard and Horan (*1998*) recognize two types of virtual communities: physical- and interest-based. But what these authors fail to recognize is that there may be situations where hybrid virtual communities will be established — communities having both physical and interest-based characteristics — and this is probable in situations of environmental management and policy research. Consider our efforts to generate open-source research for the Chesapeake Bay watershed. Some participants will have physical-based interests in smaller geographic areas falling somewhere within the broader watershed boundary (e.g., subwatersheds, cities, and towns). Other participants will have an interest in human-environment dynamics across the entire Chesapeake Bay watershed (e.g., public officials working for national level environmental agencies; marine biologists trying to understand what human activities are affecting water quality in the Bay). Still others may be less interested in the empirical context of the Chesapeake Bay, but will have a tremendous interest in theoretical and conceptual issues related to modeling of human action over space and time (e.g., researchers who are looking to do a similar modeling effort in the Adirondack Park, or researchers interested in more of the theoretical issues related to human-environment modeling and less interest in the particular setting). This eclectic group is the kind of virtual community we imagine for this endeavor. We must design the web system to enhance open-source environmental research with both interest- and physical-based researchers in mind.

Probably the biggest challenge and most critical component of this project is to develop an active virtual community filled with participants eager to collaborate. In the Linux example there are currently thousands of programmers and Unix users who are interested in the operating system. Given that there are six billion people in this world, it is not surprising that there might be many programmers who are willing to donate time and effort to the Linux project. The question is, can we generate a similar collaborative community in a research setting? Are there similar numbers of people who are interested in questions related to human dimensions of environmental change and who also possess the skills required to build an open-source modeling system? This is an open question, but certainly the user base is there — potentially every community in the world. The major question is whether we can spur the interest of enough people with the technical and theoretical interests and skills to participate. This leads us to consider individual motivation and social capital in the context of open-source research.

Individual Motivations to Collaborate

The motivation for programmers to participate in the Linux virtual community is driven largely by their desire to contribute to the development of a new and exciting technology and to gain prestige within the Linux virtual community. (10) Torvalds himself has stated that because these participants earn money elsewhere, there is no concern or need for financial gain (*Raymond, 1999*).

In academic research settings, motivations can be very similar. Researchers, especially tenured ones, have jobs with regular paychecks, so most will be financially secure. And researchers, too, are motivated to gain acceptance and prestige among their peers. In fact, the motivation for recognition by peers may be higher in academic research settings than for programmers, for promotions are determined in part by numbers of articles in peer-reviewed outlets and external letters of recognition in the academic's file. These strong motivations give us reason to believe that there is a good chance of securing well-established academic researchers to participate in our open-source research endeavor. It may, however, be more challenging to get participation from junior faculty who have not yet been tenured. The "publish or perish" incentive is a strong disincentive not to participate. Why would any junior faculty want to freely give away data they worked very

hard to collect or programs they have written before they have had a chance to publish results from them?

We have four partial solutions to this problem. First, we are not requiring people to post data or other products directly on our server. Junior faculty can simply submit a metadata record about their data or product to let people know that it exists. Other community members interested in this data or other product would still have to contact the owner of the data directly. This is actually a positive incentive for junior faculty, because it makes them known to the virtual community of researchers with similar interests (either theoretical or geographical). Second, with the "register" function, we allow researchers to tell others about their interests and expertise. We are hopeful that having that information available on a searchable web database will forge new working relationships. Third, and perhaps most importantly, we are designing our web site as an electronic peer-reviewed journal. We will have an editor (or perhaps multiple editors) and everything posted to the web site will be reviewed before being added to the searchable web-database. This review system is needed for the practical reason of server security (e.g., making sure a computer virus wasn't posted) but also to ensure that high-quality data or other research products are being submitted to the system. By conceptualizing the system as an electronic journal, participants can make references to what they post (e.g., web reviews, publications, even data or models) in their curriculum vitae. This incentive is especially vital for getting junior faculty to participate.

The fourth solution is one that is largely out of our control, but we can take steps to move the research community in the right direction. What are typically considered publications, in the academic sense, are books and journal articles. Data and computer programs are not. To some degree this is a residual from the days prior to computing and the Internet. But some academics are realizing that in this day of computing, giving away intellectual property in the form of programming code should be treated at the same level as publishing a paper (*Kiernan, 1999*). Our hope is that this system will encourage the research community to appreciate programming as another form of intellectual property that should be shared.

Academic researchers are not the only participants we are targeting as potential collaborators. There are many professional researchers in

government agencies (like the researchers at the U.S. Forest Service), nonprofit organizations (e.g., the Nature Conservancy) or even private firms that we hope will be willing to collaborate. While many may be interested in protecting their data and their turf, we think there will be many who will be motivated to participate. For example, as part of their performance review, USDA Forest Service researchers are evaluated on how useful data and reports they created were to others, including the general public. Posting data to this server and making it more easily obtained by others outside the Forest Service, will help these researchers fulfill an important job requirement. In other instances where researchers do not face an organizational "data-sharing" incentive, they still may be willing to participate in limited ways, such as in the posting of information about themselves or their organization to advertise their skills to others.

By designing the site as a new type of electronic journal, and treating the posting of data, models, publications, and web reviews as publications to this e-journal, we hope to provide enough incentive to encourage active and interested participation of actors from many disciplines. The challenge will be getting enough interest from those actors with theoretical and/or technical knowledge who can and are willing to contribute to a collaborative human-environment research endeavor that extends beyond traditional organizational boundaries. There is certainly enough general interest in human dimensions of landcover research and modeling (e.g., local and regional policy-makers, environmental groups, and even developers) that we are not worried about finding a large, interested, and motivated user community.

Building Social Capital

Building components of social capital, such as networks and trust, is something we cannot really do up front. Rather, it is something that we hope the virtual community that uses our system will gain over time. However, trust is a major concern for us, initially in the context of web server security. We need to protect the system from a user who somehow, knowingly or unknowingly, corrupts the database or server by posting something destructive like a virus, or bogus metadata records. Consequently, we will require anyone who wishes to post data to the web site to first register as a formal user of the site. Upon registering, the user will obtain a user identification and password that will allow him or her to

get access to the submit data functions. In addition, a second level of protection will be maintained by physically storing the post metadatabase and the search metadatabase as separate entities residing on physically different servers. Using our administrator computer programs, a data manager or editor will be responsible for periodic (once a week perhaps) review of metadata records and datasets that have been submitted, checking them for viruses, validity and quality issues, removing any problematic data, and then moving the accepted data over to the search metadatabase.

Establishing Conventions and Norms

We expect that, just as in the context of Linux, conventions and norms will be developed over time as participation increases. However, the literature we have reviewed on the concept of open-source emphasizes the importance of modularity in open-source initiatives and we have remained cognizant of this in both system design and as we move toward the development of open-source modeling efforts. Modularity is important to open-source endeavors so that interested participants can focus on a particular area or application of interest or, in our context, apply some element of open-source research to their own empirical setting.

Modularity in the metadata context means that we need ways of recording how metadata fits into the broader context of human-environment research. Establishing sets of common keywords in the metadata input process is one way to do this. For instance, any metadata record that is related to the Chesapeake Bay Watershed should have that phrase in its keywords entry in the metadata record. End users who are interested in the Chesapeake Bay watershed could then search the metadatabase for this phrase to find applicable metadata records. This concept of modularity should work for theoretical interests as well as geographic interests.

Modularity will be extremely important as we proceed toward the development of human-environment models. Various researchers will bring various approaches and skills to the endeavor and the system will be designed to handle that. For instance, one subgroup of participants may be interested in statistical modeling, whereas another group might be interested in spatially and temporally explicit modeling using geographic information systems. These different approaches can be thought of as different modules of a broader modeling system. By being cognizant of many modeling

approaches (e.g., Markov chain, logistic function, regression, geographic information system, ecosystem simulation, agent-based, etc.) the open-source research community may be able to make further advances in how to possibly link several approaches in new and creative ways.

Another important standard that will probably evolve over time is one related to geographic information systems data. One common problem often encountered with GIS data is incompatibility because of different map projections or datums used in generating the dataset and different standards related to georeferencing accuracy. For instance, U.S. state agencies typically use the State Plane projection while U.S. Federal agencies often use the Universal Transverse Mercator (UTM) projection. As the metadatabase and server acquires data, standards will need to be developed through discussions with open-source research system collaborators. This will be critical to ensure data compatibility and to provide the ability to "scale up" geographically from an analysis at a sub-watershed level to the full watershed area.

Establishing Rules of Participation

We have already mentioned the submit data rule that will be established that requires users to be formally registered with us prior to submitting data or metadata. On the other hand, we have decided that the "search for data" mechanism will have no such requirement, and will be available to anyone who has access to the web browser and the Internet.

But the most important rules for open-source software endeavors, such as Linux, are the licensing rules, and in an open-source research context we need to think about these as well. O'Reilly and Dyson (*1998*) provide an overview of open-source licensing so we will not repeat that here. One of the crucial design issues to our web system is establishing our license agreement and making it clearly visible. We are planning to follow the "GNU general public license" (GPL) approach and the "Copyleft" principle, where users are granted permission to download data and/or the source code of modeling programs. In any analysis that is produced, we will require users to cite the original creator of the data or program and also to identify where they acquired these products. And like the GPL approach in the context of Linux, users are obligated to distribute, without fee or additional license terms, the data and the source code of all derivative

works. We of course hope any enhancements made will be posted back to our web metadatabase as a new version so advances continually will be made.

Conclusion

This paper described our efforts to extend the concept of open-source programming into the realm of scientific research and specifically to address the complex problem of human-environment modeling. We join a small but growing group of researchers who see the open-source programming revolution as a natural extension of scientific endeavors (*O'Reilly and Dyson, 1998; Gezelter, 1999, 2000*). We are currently in beta-test of release 1.0 of the system and will be fully online by the spring of 2001. The existing system can be visited at www.open-research.org. Interested readers are encouraged to register with the system.

We strongly believe that the concept of an open-source approach, coupled with recent advancements in the development of virtual communities via the Web, has tremendous potential to improve our ability to understand complex systems. To paraphrase Raymond (*1999*), the more eyes studying a complex problem the more likely solutions will be discovered. In the context of human-environmental research, the first step toward developing an open-source endeavor is through the development of a web-metadatabase system and peer reviewed e-journal such as we describe. As Gezelter (*1999, 2000*) argues, this is a natural extension to scientific endeavors in that the open sharing of publications, data, and programs (models) are all part of the tradition of verifiable science. Our vision of web-based collaboration strives to encourage this concept of open sharing. Further, our effort strives to ignite a new level of creative and collective problem-solving of a very complex problem — land use modeling research. Central to this effort is the question of how to effectively motivate academic and nonacademic researchers alike from a variety of disciplines, organizations, and countries to participate. This paper was written, in part, to understand those challenges to scientific participation. But if we can identify and effectively implement incentives for participation, and get these incentives correct, the "Open Research System" we are developing — a new kind of web-based peer-reviewed journal, supporting a metadata repository with user-friendly search mechanisms, and modular open-source modeling initiatives — will move us

beyond the traditional methods of scientific research to what possibly may become a new research paradigm with the promise of achieving new levels of productivity, discourse, and truly global, collaborative problem solving. To tackle complex problems like human dimensions of global change issues, this is what we need.

Notes

1. See for example, NSF's Human Dimensions of Global Change program at http://www.nsf.gov/sbe/hdgc/hdgc.htm, the Biocomplexity of the Environment program at http://www.nsf.gov/home/crssprgm/be/ or the Long Term Ecological Research program at http://lternet.edu.

2. Commercial software is usually made available in compiled and unreadable form—in binary (1's and 0's) that make sense to computer microprocessors but are unreadable to programmers. By posting the source code on the Internet, Torvalds made his intellectual property, the Linux logic, readable to other programmers.

3. See, for example, the section entitled "Sizing Up the Open-Source Community" in the document http://www.edventure.com/release1/1198.html.

4. For example, the Internet browser company Netscape has now made their browser software open-source hoping to capitalize on the available creative powers "out there" in cyberspace (*Opensource.org, 2000*). Another open-source initiative is "Prospero," an interlibrary loan software package which is in use by over 100 institutions (*Kiernan, 1999*).

5. Anyone who has taken a course on operating system design and development would argue that the task that Torvalds and others took on--to program from scratch an operating system--is indeed a complex task.

6. Human capital can be defined as the "acquired knowledge and skills that an individual brings to an activity" (*Ostrom, 1999a: 175*).

7. The term "metadata" refers to data about data. Good metadata for a dataset will usually describe when the data was collected or produced, who produced it, what are the limitations in the data (e.g., in a spatial dataset it may have information on the resolution of the data), what time point the data represents, etc. When combining data from various sources for some analysis, metadata is crucial to make sure the data are compatible.

8. For example, the University of Michigan's Inter-university Consortium for Political and Social Research (ICPSR) at http://www.icpsr.umich.edu/ or the Federal Geographic Data Committee Clearinghouses (*http://www.fgdc.gov/clearinghouse/clearinghouse.html*).

9. These participants include CIPEC at Indiana University, the Mid-Atlantic Integrated Assessment program at the US Environmental Protection Agency, researchers at Paul Smith's College in the Adirondack Park region, researchers at the University of Massachusetts, Amherst, researchers at the Northeastern Research Station of the USDA Forest Service, Burlington Vermont, and just recently, researchers studying the Delaware River basin.

10. A dislike of the dominance of Microsoft appears to be another motivating factor for some Linux programmers.

References

- Chesapeake Bay Program, *(*2000). 20 March. <http://www.chesapeakebay.net/wshed_intro.htm>
- Fountain, J.E. (1997). Social Capital: A Key Enabler of Innovation in Science and Technology, In L.M. Branscomb and J. Keller, eds. Investing in Innovation: Toward a Consensus Strategy for Federal Technology Policy. Cambridge: MIT Press.
- Blanchard, A. and Horan, T. (1998). Virtual Communities and Social Capital, Social Science Computer Review, 16 (3): 293-307.

- Gezelter, J.D. (1999). Catalyzing Open Source Development in Science: The Open Science Project, 19 Mar 2000, <http://www.openscience.org/talks/bnl/OSOS.pdf>

- Gezelter, J.D. (2000). 3 March 2000 <http://www.openscience.org>

- Kiernan, V. (1999). The 'Open Source Movement' Turns Its Eye to Science, The Chronicle of Higher Education, November 5: A51

- Learmonth, M. (1997). Giving It All Away, 15 Feb 2000, <http://www.metroactive.com/papers/metro/05.08.97/cover/linus-9719.html>

- McHugh, J. (1998). For the Love of Hacking, Forbes, 20 Mar 2000, <http://www.forbes.com/forbes/98/0810/6203094a.htm>

- Opensource.org, (2000). Frequently Asked Questions About Open Source, 15 Feb 2000, <http://www.opensource.org/faq.html>

- O'Reilly, T. (1999). Lessons from Open-Source Software Development, Communications of the ACM, 42 (4): 33.

- O'Reilly, T. and Dyson, E. (1998). The Open-Source Revolution, Release 1.0: Ester Dyson's Monthly Report. 14 Mar 2000, <http://www.edventure.com/release1/1198.html>

- Ostrom, E. (1990). Governing the Commons: The Evolution of Institutions for Collective Action. New York: Cambridge University Press.

- Ostrom, E. (1992). Crafting Institutions for Self-Governing Irrigation Systems. San Francisco, CA: ICS Press.

- Ostrom, E. (1998). A Behavioral Approach to the Rational Choice Theory of Collective Action. American Political Science Review 92(1) (March): 1-22.

- Ostrom, E. (1999a). Social Capital: A Fad or a Fundamental Concept? In Partha Dasgupta and Ismail Seraeldin, eds. Social Capital: A Multifaceted Perspective. Washington D.C: The World Bank: 172-214.

- Ostrom, E. (1999b). Self-Governance and Forest Resources, Center for International Forestry Research (CIFOR) Occasional Paper No. 20. February.

- Putnam, R., Leonardi, R., and Nanetti, N. (1993). Making Democracy Work: Civic Traditions in Modern Italy. Princeton, New Jersey: Princeton University Press.

- Putnam, R.D. (1995a). Bowling Alone: America's Declining Social Capital, Journal of Democracy, 6: 65-78.

- Putnam, R.D. (1995b). Tuning In, Tuning Out: The Strange Disappearance of Social Capital in America, Political Science and Politics, 28: 664-483.

- Raymond, E. (1999). The Cathedral and the Bazaar, 16 Mar 2000, <http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar.html>

- Raymond, E. (2000). "How to Become a Hacker," 24 Mar 2000 <http://www.tuxedo.org/~esr/faqs/hacker-howto.html>

- Rheingold, H. (1993). The Virtual Community: Homesteading on the Electronic Frontier. New York: Harper Perennial.

- The Economist. (1998). Red Hat Trick: Linux Operating System May Pose a Serious Threat to Microsoft Dominance, The Economist. 348 (8088): 76. Oct 3.

- USDA Forest Service Northern Global Change Research Program, (2000). Carbon Budget of United States Forests, 20 Mar 2000, <http://www.fs.fed.us/ne/global/research/carbon/forcarb.html>

- Wallich, P. (1999). The Best Things in Cyberspace are Free, Scientific American, 280 (3): 44.

- Wilson, G. (1999). A Natural Home for Open Source: A Report on Open Source / Open Science 99, 5 Mar 2000, <http://www.ddj.com/articles/1999/9975/9975q/9975q.htm>